# 3530 D & F
# COMMUNICATING WITH ORION *delta*

## OPERATING MANUAL

Solartron Instruments

OFFICIAL RECORDS STATION
FILE COPY
FACILITIES JAN 12 1987

MP MASTER
MA20217
SOLARTRON
Schlumberger

# 3530 D & F
# COMMUNICATING WITH ORION *delta*
## OPERATING MANUAL

Issue 1, May 1984

Part No. 35302042

SOLARTRON
Schlumberger

Solartron Instruments, Victoria Road, Farnborough
Hampshire, England GU14 7PW Telephone: Farnborough (0252) 544433
Telex: 858245 Solfar G Cables: Solartron Farnborough
A division of Schlumberger Electronics (UK) Ltd

Solartron pursues a policy of continuous development and product improvement
The specification in this document may therefore be changed without notice

19©84

# Contents


| | Para. No. | Title | Page No. |
|---|---|---|---|
| **Chapter 1** | **Introduction** | | |
| | 1 | Introduction | 1.2 |
| | 2 | ORION Delta Command Language | 1.2 |
| | 3 | ORION Delta Programming Language | 1.2 |
| | 4 | ORION Delta Interface Options | 1.3 |
| | 4.1 | RS232 Interface | 1.3 |
| | 4.2 | Fast Serial RS422/232 (BS4505) | 1.3 |
| | 4.3 | GPIB IEE 488 | 1.3 |
| **Chapter 2** | **ORION Command Language** | | |
| | 1 | Introduction | 2.2 |
| | 2 | Command Syntax | 2.2 |
| | 2.1 | Sending commands to ORION | 2.3 |
| | 3 | Channel Sets | 2.4 |
| | 4 | Rules for entering ORION commands | 2.5 |
| | 5 | Command string examples | 2.6 |
| | 5.1 | Example 1 Channel definition | 2.6 |
| | 5.2 | Example 2 Measurement Task | 2.6 |
| | 5.3 | Example 3 Measurement Task to count events | 2.8 |
| | 5.4 | Example 4 Event Task | 2.9 |
| | 5.5 | Example 5 Attach an analogue output channel to a measurement channel | 2.10 |
| | 5.6 | Example 6 Measurement Task with end of scan processing | 2.10 |
| **Chapter 3** | **Channel Definition** | | |
| | 1 | Introduction | 3.2 |
| | 1.1 | Channel Syntax | 3.2 |
| | 2 | Analogue Input or 'Decimal' type Digital Input | 3.3 |
| | 3 | Analogue Output | 3.5 |
| | 4 | Status type digital Input | 3.6 |
| | 5 | Event type digital Input | 3.8 |
| **Chapter 4** | **Task Definition** | | |
| | 1 | Introduction | 4.2 |
| | 1.1 | Task Syntax | 4.2 |
| | 2 | Measurement task | 4.3 |
| | 3 | Event task | 4.9 |

| *Para. No.* | *Title* | *Page No.* |
|---|---|---|
| **Chapter 5** | **Control and other Commands** | |
| 1 | Control Commands | 5.2 |
| 1.1 | RUn | 5.2 |
| 1.2 | HAlt | 5.2 |
| 1.3 | PAuse | 5.2 |
| 1.4 | GO | 5.2 |
| 1.5 | Cross trigger | 5.2 |
| 1.6 | Cross Abort | 5.2 |
| 2 | Initialisation command | 5.3 |
| 3 | Cartridge command | 5.4 |
| 4 | Monitor command | 5.6 |
| 5 | Measure command | 5.7 |
| 6 | List command | 5.8 |
| 7 | Renumber command | 5.9 |
| 8 | Drift command | 5.9 |
| 9 | Set command | 5.9 |
| 10 | Write message command | 5.10 |
| 11 | Conversion Definition command | 5.11 |
| 12 | Text Definition command | 5.12 |
| 13 | Query command | 5.12 |
| 13.1 | Querying a parameter | 5.12 |
| 13.2 | Query only commands | 5.13 |
| **Chapter 6** | **The ORION Processing Program** | |
| 1 | Introduction | 6.2 |
| 2 | Processing Facilities | 6.2 |
| 3 | The Program Structure | 6.3 |
| 3.1 | Line numbers | 6.3 |
| 3.2 | Statements | 6.3 |
| 3.3 | Statement separators (:) | 6.4 |
| 4 | Creating a program | 6.5 |
| 4.1 | Entering a program | 6.5 |
| 4.2 | Correcting a line | 6.5 |
| 4.3 | Deleting a line | 6.5 |
| 4.4 | Inserting a line | 6.5 |
| 4.5 | Renumbering program lines | 6.6 |
| 4.6 | List a program | 6.6 |
| 5 | Program constants | 6.7 |
| 6 | Program variables | 6.7 |
| 6.1 | Predefined variables | 6.7 |
| 6.2 | User defined variables | 6.10 |
| 7 | Mathematical operators and functions | 6.10 |
| 8 | Control Structures | 6.11 |
| 8.1 | Decision making – IF THEN ELSE ENDIF | 6.11 |
| 8.2 | Program loops – FOR . . . NEXT | 6.12 |
| 8.3 | Unconditional branch – GOTO | 6.13 |
| 9 | The Output Statements | 6.14 |
| 9.1 | The PRINT statement | 6.14 |
| 9.2 | PRINT statement extensions | 6.16 |
| 9.3 | Output format declaration | 6.17 |
| 9.4 | The WRITE statement | 6.18 |

| Para. No. | Title | Page No. |
|---|---|---|
| **Chapter 6** *continued* | | |
| 9.5 | The KILL statement | 6.20 |
| 9.6 | SET . . . TO statements | 6.20 |
| 10 | Digital Inputs | 6.21 |
| 11 | Control of other Tasks | 6.22 |
| 11.1 | Triggering other Tasks | 6.22 |
| 11.2 | Aborting other Tasks | 6.22 |
| 12 | Counters and Timers | 6.22 |
| 12.1 | Counters COUNT1 to COUNT8 | 6.22 |
| 12.2 | Timers TIMER0 to TIMER9 | 6.22 |
| 13 | Random Channel Access | 6.23 |
| 14 | Memory Capacity | 6.24 |
| 14.1 | Processing Program Memory Requirements | 6.24 |
| 14.2 | Statement Memory Allocation | 6.25 |
| 15 | Speed of Execution | 6.26 |
| 16 | Programming Techniques | 6.26 |
| 17 | Vocabulary of Functions | 6.28 |
| **Chapter 7** | **Output and Recording Formats** | |
| 1 | General | 7.2 |
| 2 | Output Data Formats | 7.2 |
| 2.1 | Numerical Result | 7.3 |
| 2.2 | Start and End of Scan Markers | 7.3 |
| 2.3 | Event Result | 7.4 |
| 2.4 | Run and Halt Markers | 7.5 |
| 2.5 | Power Fail and Power Back Messages | 7.5 |
| 2.6 | Date | 7.6 |
| 2.7 | Specific Formatting | 7.6 |
| 3 | Coded Binary Formatting | 7.8 |
| 3.1 | Numerical Result | 7.11 |
| 3.2 | Start and End of Scan Markers | 7.11 |
| 3.3 | Event Result | 7.12 |
| 3.4 | Run and Halt Markers | 7.12 |
| 3.5 | Power Fail and Power Back | 7.13 |
| 3.6 | Date | 7.13 |
| 4 | Recording Data on Cartridge (3530D only) | 7.13 |
| 4.1 | Recording Modes | 7.13 |
| 4.2 | Recording Formats | 7.14 |
| 4.3 | Data Output Format | 7.14 |
| 4.4 | ORION Tape Program | 7.15 |
| 4.5 | The Cartridge Tape Format | |
| **Chapter 8** | **Local Terminal Interface** | |
| 1 | Introduction | 8.2 |
| 2 | Local Terminal Protocol | 8.2 |
| 3 | Protocol Summary | 8.3 |
| 4 | Connecting the Terminal | 8.4 |

| Para. No. | Title | Page No. |
|---|---|---|
| **Chapter 9** | **Standard RS232 Interface** | |
| 1 | Introduction | 9.2 |
| 2 | Operating with a computer using the Standard RS232 Interface | 9.3 |
| 2.1 | Result Transfer Mode | 9.3 |
| 2.2 | Control Mode | 9.4 |
| 2.3 | Communication using simple Protocol | 9.7 |
| 3 | Operating with a VDU or Printing Terminal | 9.11 |
| 3.1 | Result Transfer Mode | 9.11 |
| 3.2 | Control Mode | 9.11 |
| 3.3 | Relinquishing control | 9.11 |
| 4 | Output Only protocol | 9.11 |
| 5 | Connecting to Standard RS232 Port | 9.12 |
| **Chapter 10** | **Fast Serial Interface** | |
| 1 | Introduction | 10.2 |
| 2 | RS422/232 Operation with BS4505 Protocol | 10.2 |
| 3 | Other RS232 Protocol Available | 10.7 |
| 4 | Serial Interface Connection Details | 10.8 |
| 4.1 | Cable for RS422 Operation | 10.9 |
| **Chapter 11** | **GPIB Interface (IEEE 488)** | |
| 1 | Introduction | 11.2 |
| 2 | GPIB status | 11.2 |
| 2.1 | Talker only status | 11.2 |
| 2.2 | Talker/Listener status | 11.2 |
| 3 | GPIB protocol | 11.3 |
| 4 | Service Request | 11.4 |
| 5 | Polling | 11.5 |
| 5.1 | Parallel Poll | 11.5 |
| 5.2 | Serial Poll | 11.5 |
| 6 | Input/Output data | 11.5 |
| 6.1 | Input data | 11.5 |
| 6.2 | Output data | 11.5 |
| 7 | The GPIB address | 11.5 |
| 8 | Communication via the GPIB bus | 11.6 |
| 9 | GPIB interface connection details | 11.7 |
| **Appendix 1** | **Error Messages** | |
| 1 | Programming Errors and Other Faults indicated on the display | A1.1 |
| 2 | Measurement Errors included in the Output results | A1.3 |
| 3 | Tape Handling Message | A1.4 |

**Appendix 2 Sensor Number Tables**

**Appendix 3 Example Program – Orion/HP85 RS232 Interface**

| | | |
|---|---|---|
| 1 | Hardware | A3.4 |
| 2 | Operating Instructions | A3.4 |
| 3 | Example Program Breakdown | A3.6 |

**Appendix 4 Example Program – Orion/HP85 via GPIB**

| | | |
|---|---|---|
| 1 | Hardware | A4.2 |
| 2 | Operating Instructions | A4.2 |
| 3 | Example Program Breakdown | A4.3 |

Digitized by the Internet Archive
in 2025

https://archive.org/details/manualsplus_07234

# Chapter 1
# Introduction


| *Para. No.* | *Title* | *Page No.* |
|---|---|---|
| 1 | Introduction | 1.2 |
| 2 | ORION Delta Command Language | 1.2 |
| 3 | ORION Delta Programming Language | 1.2 |
| 4 | ORION Delta Interface Options | 1.3 |
| 4.1 | RS232 Interface | 1.3 |
| 4.2 | Fast Serial RS422/232 (BS4505) | 1.3 |
| 4.3 | GPIB IEE 488 | 1.3 |

# 1. INTRODUCTION

The ORION delta is a powerful data logger that not only boasts a comprehensive and flexible logging capability but also offers a powerful data processing facility. ORION can be set up and controlled either locally, from the front panel, or remotely using a computer or other intelligent peripheral connected to one of the remote interfaces. An RS232 interface is supplied as standard, RS422 and GPIB IEEE488 are available as options.

Two versions of the ORION delta data logger are available:-

**3530D –** **ORION delta;** has a front panel keyboard and display. Setting up may be performed using the keyboard in conjunction with the display which offers menus from which the operator may make the required selection. A printer and a cartridge recorder are built in.

**3530F –** **ORION F;** is a 'Blind' ORION which has no front panel controls or displays, neither printer nor recorder are included. It has, however, the same logging capabilities and processing facilities as the ORION delta.

**Note:** The general term 'ORION' has been used, throughout the manual when discussing functions available to both versions of data logger. However, when describing functions that are specific to a single version logger, the term 3530D or 3530F is used, as appropriate.

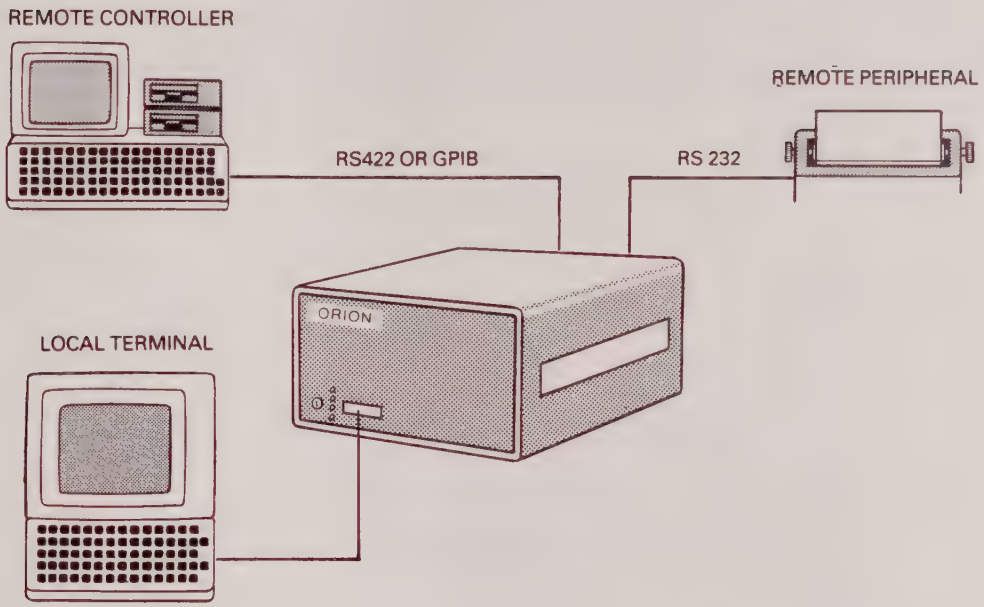
# 2. ORION DELTA COMMAND LANGUAGE

The ORION command language is based on simple English words and is both flexible and easy to use, allowing channel information to be entered and tasks to be set up and run with the minimum of effort. Up to eight tasks may be set up to run singularly or concurrently as required.

# 3. ORION DELTA PROGRAMMING LANGUAGE

Associated with each task there may be, if required, a user written processing program. The programs are written using a BASIC like language and can, for example, be used to act on the measured results for the associated task when the scan task has finished but before the results are output. This offers an unlimited number of possibilities for obtaining derived results from the measured channels. Some other examples of processing program uses are: data reduction, Task cross triggering or aborting, output control using analogue or digital signals.

# 4. ORION DELTA INTERFACE OPTIONS




The ORION can be controlled locally, using the Front panel keys of an ORION delta, 3530D, or via a terminal attached to the local interface socket of an ORION F, 3530F, using the command language.

The ORION may also be controlled remotely, using a computer, or other intelligent peripherals, connected to the standard or one of the optional interface sockets located on the ORION back panel.

## 4.1 RS232 INTERFACE

An RS232 socket on the controller board is supplied as standard and is suitable for connection to a computer, or any device capable of controlling the ORION via a RS232 link. In addition the socket may be used to connect an output only recording device, such as a printer or punch.

## 4.2 FAST SERIAL, RS422/232 (BS4505)

The Fast serial interface board, 35304B, is an optional board having two output ports that may be set independently for either RS422 or RS232 operation, with BS4505 protocol. In addition limited RS232 protocol is available.

## 4.3 GPIB IEE(488)

The GPIB interface board, 35804A, fully implements the GPIB IEEE488 (1978) interface standard and will support any GPIB compatible device such as a computer, calculator or output peripheral. ORION may be set up to be a Talker/listener device or Talker only device.

# Chapter 2
# Orion Command Language


| *Para. No.* | *Title* | *Page No.* |
|---|---|---|
| 1 | Introduction | 2.2 |
| 2 | Command Syntax | 2.2 |
| 2.1 | Sending commands to ORION | 2.3 |
| 3 | Channel Sets | 2.4 |
| 4 | Rules for entering ORION commands | 2.5 |
| 5 | Command string examples | 2.6 |
| 5.1 | Example 1 Channel definition | 2.6 |
| 5.2 | Example 2 Measurement Task | 2.6 |
| 5.3 | Example 3 Measurement Task to count events | 2.8 |
| 5.4 | Example 4 Event Task | 2.9 |
| 5.5 | Example 5 Attach an analogue output channel to a measurement channel | 2.10 |
| 5.6 | Example 6 Measurement Task with end of scan processing | 2.10 |

## 1. INTRODUCTION

The ORION Command language is both versatile and simple to use. When controlling ORION from a remote interface it provides the means by which the ORION database may be set up, interrogated or modified. In addition it provides the user with the capability of running, halting and aborting tasks or monitoring or taking measurements on specific ORION channels.

The ORION command language can be divided into three categories.

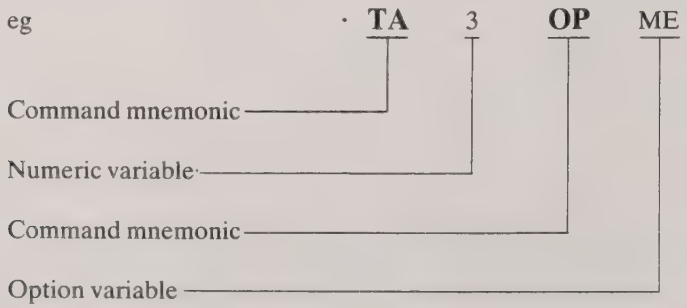
| | |
|---|---|
| CONTROL commands: | run, halt, etc.<br>initialisation<br>cartridge tape command |
| DEFINITION commands: | channel definition<br>task definition<br>conversion definition<br>text definition |
| OTHER commands: | monitor<br>measure<br>list<br>miscellaneous commands |

In addition to the features described above, when defining a task to run on ORION, the user may select the processing facility as an extension of the task being defined. This facility allows the user to extend the processing capability of the task and is described in Chapter 6.

## 2. COMMAND SYNTAX

In general an ORION command consists of a command word followed by a parameter. Virtually every ORION command is an English word of which the first two characters are recognised by ORION as the command mnemonic. The user may type the two character mnemonic or use the whole or part of the command word, eg CH, CHAN, CHANNEL are all valid forms of the same command.

In addition to the command mnemonic most commands require the addition of a qualifying parameter. This may be a further two character mnemonic selected from a list of options or a numeric or alpha numeric variable.



In this manual, for the purpose of clarity, the following conventions have been used when displaying examples of command syntax.

1. Command mnemonics are shown in large **BOLD** capitals.

2. Option variables are shown in small LIGHT CAPITALS.

3. Numeric variables are shown in *lower case italics*.

4. Alpha strings are shown in lower case alpha.

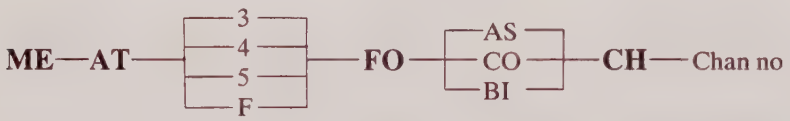
Chapters 3 to 5 describe the ORION commands, related commands have been grouped together to give a tree like structure known as the command tree. The command trees define all the possible options associated with a specific function eg. definition of Analog output channels, definition of an event task, single channel measurements, etc. However, many of the commands shown in that tree are optional, and the user need only enter the options required.

## 2.1 SENDING COMMANDS TO ORION

When a command or a group of commands are entered they are referred to as the **command string.** The command string sent to ORION must start with the command mnemonic that defines the tree of which all subsequent commands in the string form part. The next command entered in that string may be any valid command from the command tree. Finally, when all related commands have been entered, the string must be terminated using an end of line character (C/R).

**For example:** Send a command to ORION that will take a single measurement on a specified channel.

The command tree syntax for the Measure command is:




The command mnemonic that defines the measure command tree is **ME** and must be the first command in the input command string. As can be seen from the tree the **ME** command requires no qualifying parameter. The next command to be entered may be any command from the tree.

To send a command string to ORION to instruct ORION to take a single measurement on channel 10 and output the result in full ASCII format, enter the command:

**ME FO** AS **CH**10 C/R

Providing that ORION is switched on and that channel 10 has previously been defined, a single measurement will be made on channel 10, and the result will be output to the device sending the command.

**Note:** The command **AT** has been omitted, the default value of F or the value used for any previous measure command will be used.

Should there be an eror in the command string sent to ORION, ORION will respond with an error message, see Appendix 1 for details of ORION error message.

## 3. CHANNEL SETS

Throughout this manual references are made to channel sets, the valid formats of these sets is described below:

1. 'Single channel' set: specify the single channel number required.

   eg **CH**2

2. 'Multi channel' set: specify each channel number separated by either a comma (,) or an ampersand (&). The list of channel numbers is only limited by the total input line length ie 80 characters. Channels need not be specified in ascending order.

   eg **CH**1,35,6,11,4&15&23

3. 'Range of channels' set: specify the 'start' channel number and the 'end' channel number separated by a hyphen (-).

   eg **CH**40-59

   the range may also be specified in descending order.

   eg **CH**59-40

   or mixed with 'single' or 'multi' channels.

   eg **CH**1,2,10-20,5

4. When ORION is controlling one or more slave units it may be required to define a channel set which spans more than one unit. A delimiting character must be used to separate the channels of each unit, ie channels 1 to 200 of master ORION, channels 201 to 400 and 401 to 600 for ORION slave units.

   eg **CH**101-200, 201-300

   Where channels 101 to 200 are master ORION channels and 201 to 300 are slave ORION channels.

   Channels may be entered in any numerical order, and will be scanned in the sequence in which they are entered.

## 4. RULES FOR ENTERING ORION COMMANDS

When sending a command string to ORION delta the following rules must be observed.

i) The command string must always start with the command mnemonic defining the command tree.

eg **TA,CH,MO**

ii) A command string may only contain one tree defining command.

iii) A delimiting space must be entered between a command and an alpha variable or option. A space is optional if the variable is numeric.

eg **CA MO** CY — delimiting spaces required
**CH**4 **AL**5 — delimiting spaces optional

iv) A command string must be terminated by a carriage return, C/R.

v) The total number of characters in a command string must not exceed 80 characters, including delimiting space.

**Note:** To minimise errors when entering command data the command string may be broken down into a number of smaller strings each being entered separately. However each of the small strings must also start with the command that defines the tree.

eg **TA, CH, MO**

iv) If a command and its parameter is entered more than once during the definition of a channel set, task, etc, the last value received for that parameter will be the value used.

eg **TA**1 **CH**5-10 **AT** F **PR** ON **AT** 4 C/R

where the measurement rate of 100 channels per second, **AT**4, will be used and not **AT** F, 10 channels per second.

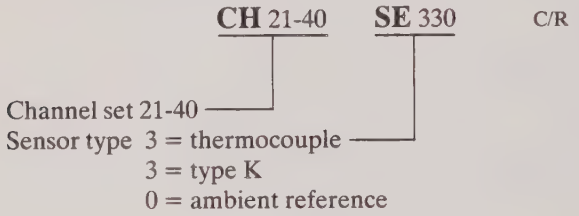
iiv) If a command string contains an illegal option the whole command string will be ignored and must be re-entered.

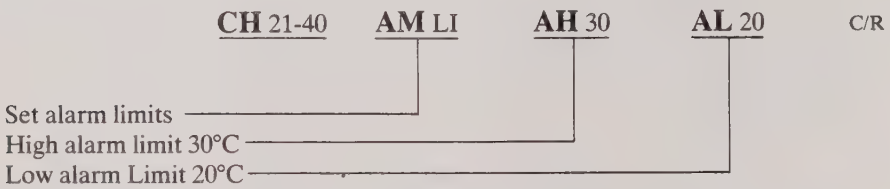
## 5. COMMAND STRING EXAMPLES

The examples given in this section are typical command strings for channel and task definition. In each example a description of the channel definition or task function is given, followed by the command string broken down into its component parts, and an accompanying explanation.

### 5.1 EXAMPLE 1 CHANNEL DEFINITION

Define channel set channels 21 to 40 inclusive, as thermocouple sensors type K with ambient thermocouple reference junction. Select an upper alarm limit of 30°C and a lower limit of 20°C.

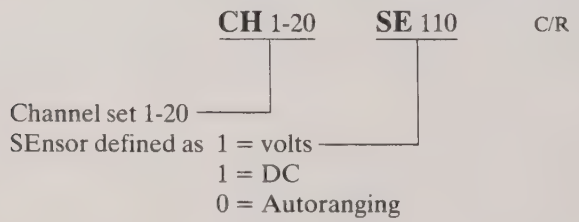


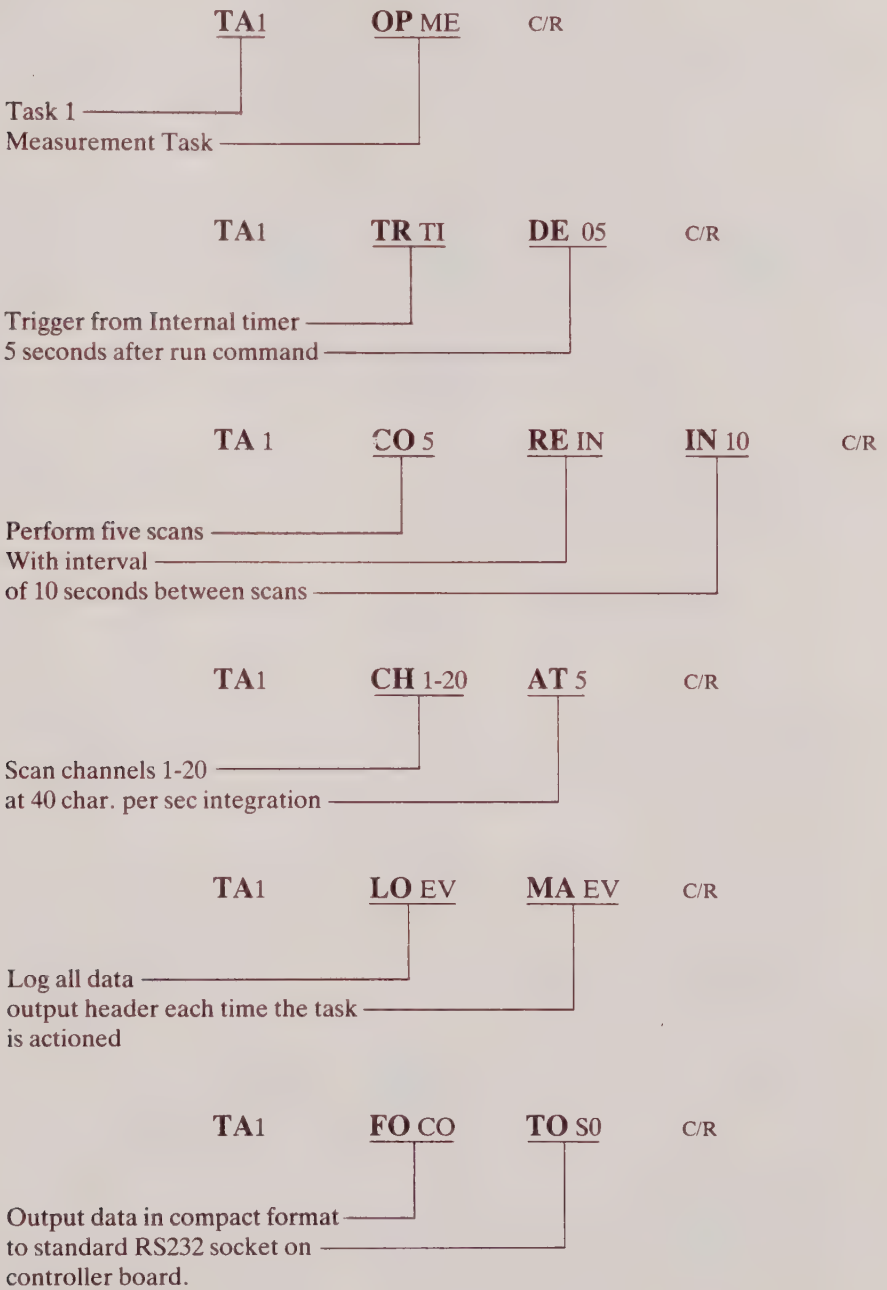

**CH** 21-40 **SE** 330 **AM** LI **AH**30 **AL**20 C/R

### 5.2 EXAMPLE 2 MEASUREMENT TASK

Define channel set for measurement task as channels 1 to 20 inclusive, measuring volts DC, using autoranging.



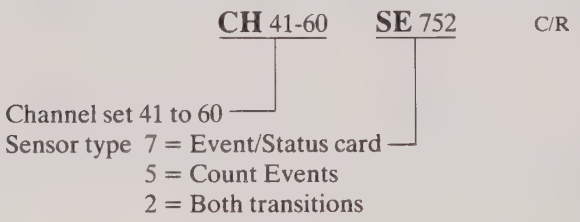
Define a measurement task to be triggered from the internal timer five seconds after the RUn command has been given. The task will perform five scans on channels 1 to 20, with a ten second interval between scans. The measurement rate will be 40 channels per second and all data will be logged. Data will be output to the standard RS232 interface on the controller board in compact ASCII format. A header marker will be output each time that the task is actioned.


TA1 OP ME C/R
Task 1
Measurement Task
TA1 TR TI DE 05 C/R
Trigger from Internal timer
5 seconds after run command
TA 1 CO 5 RE IN IN 10 C/R
Perform five scans
With interval
of 10 seconds between scans
TA1 CH 1-20 AT 5 C/R
Scan channels 1-20
at 40 char. per sec integration
TA1 LO EV MA EV C/R
Log all data
output header each time the task
is actioned
TA1 FO CO TO S0 C/R
Output data in compact format
to standard RS232 socket on
controller board.

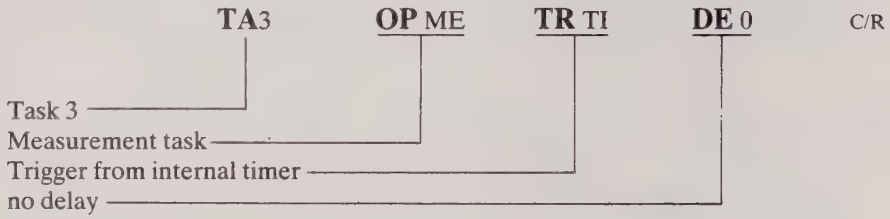
**TA1 OP ME TR TI DE05 CO 5 RE IN IN10 CH1-20 AT 5LO EV MA EV FO CO TO S0 C/R**

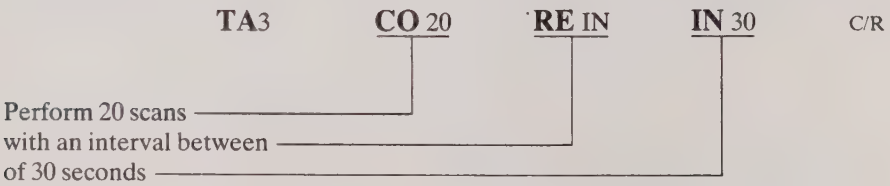
## 5.3 EXAMPLE 3 MEASUREMENT TASKS TO COUNT EVENTS

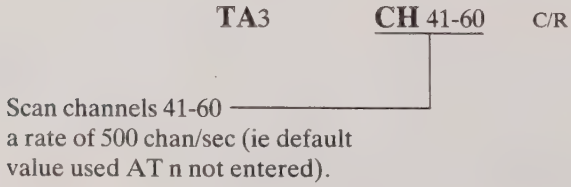
Define channel set as channels 41 to 60 of an event-status module, to count events with either transition being recognised.

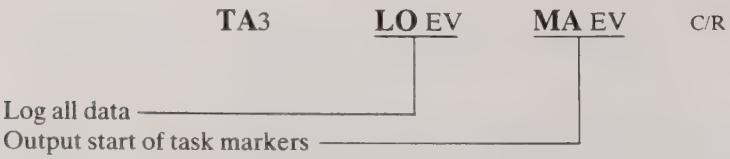


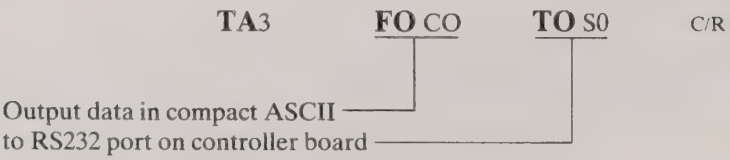

Define measurement task 3 to be triggered immediately a run command is given. Perform twenty scans on channels 41 to 60 with a 30 second interval between the start of each scan. Log all data, include start of task markers and output to RS232 port on controller board in compact ASCII format.



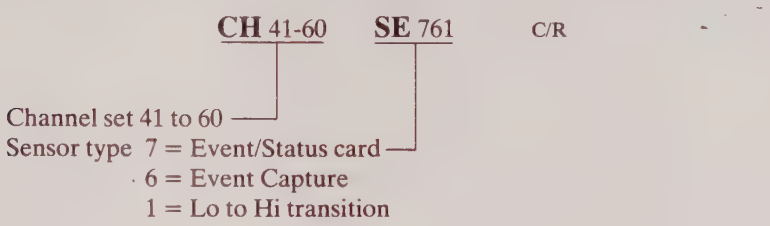



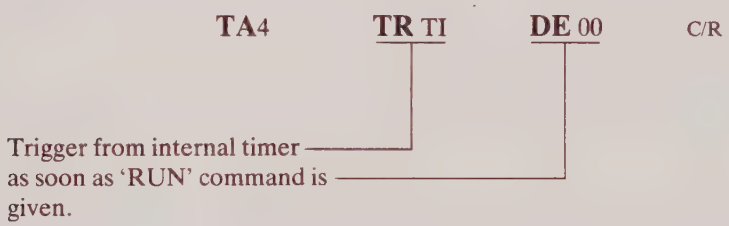

**TA3OP** ME **TR** TI **DE0CO20RE** IN **IN30 CH**41-60 **LO** EV **MA** EV **FO** CO **TO** S0 C R

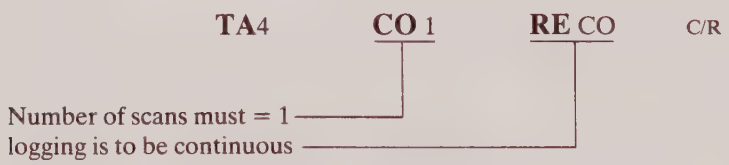
## 5.4 EXAMPLE 4 EVENT TASK

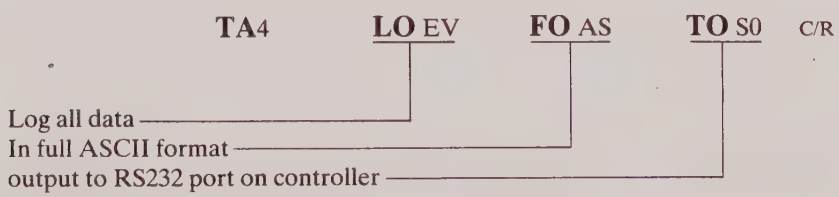
Define channels 41 to 60 of an event status module to capture events of a Lo to Hi transition.




Define task 4 as an event task triggered from the internal timer immediately the run command is given. Event monitoring is to be continuous on channels 41-60. All data is to be logged in compact ASCII to the RS232 port on the controller board.



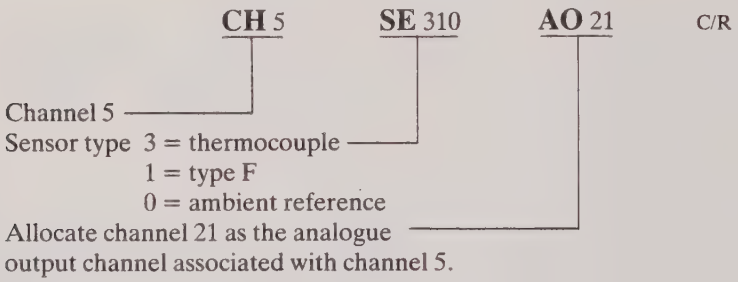





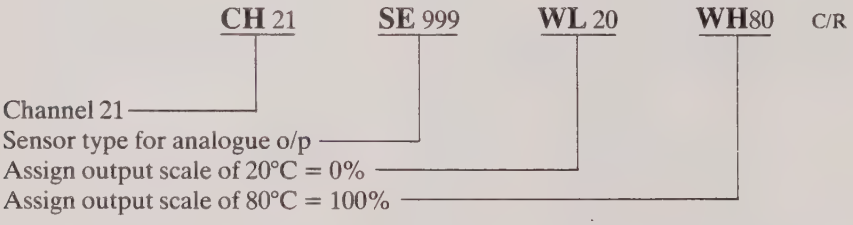

**TA4OP** EV **TR** TI **DE0CO1 RE** CO **CH**41-60 **LO** EV **FO** AS **TO** S0 C/R

## 5.5 EXAMPLE 5 ATTACH AN ANALOGUE OUTPUT CHANNEL TO A MEASUREMENT CHANNEL

Define measurement channel 5 as a temperature measurement channel and allocate analogue output channel 21 as output channel attached to it.

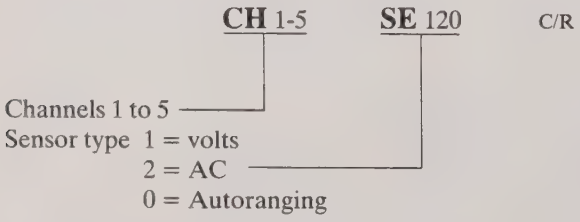


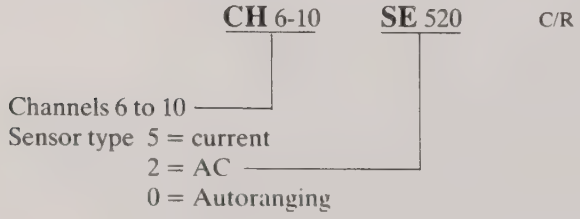

Define channel 21 as an Analogue output channel, where the output is to represent a range of 20°C to 80°C on the Measurement channel, channel 5. Therefore assign a scale value of 20°C for 0% output and 80°C for 100% output.



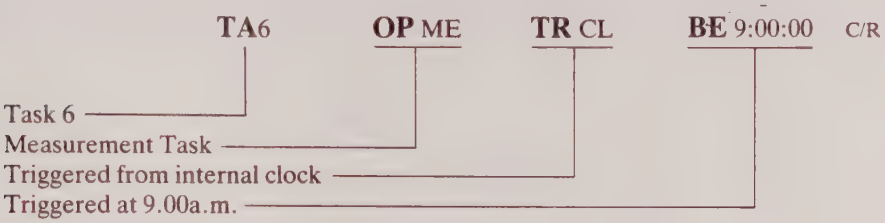

## 5.6 EXAMPLE 6 MEASUREMENT TASK WITH END OF SCAN PROCESSING

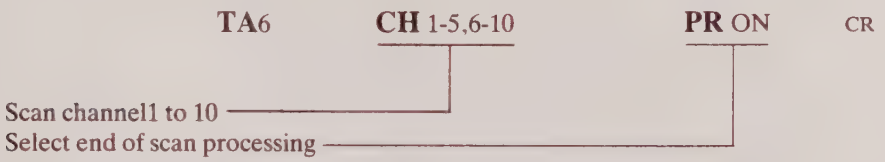
Define the channel set for the measurement task, Channels 1 to 5 measures voltage AC, channels 6 to 10 measures current AC.



Define a measurement task, to be triggered from the ORION internal 24 hour clock, to scan channels 1 to 10 continuously. End of scan processing to be selected, all data to be logged to the GPIB interface in full ASCII format.



**TA6 OP** ME **TR** CL **BE**9:00:00 **CH**1-5,6-10 **PR** ON **LO** EV **FO** AS **TO** GP C/R

Define a processing program to process the measured results from channels 1 to 5 and 6 to 10 to give the average reading over five measurements for each group.

```
6010  VOLTS = 0
6020  CURRENT = 0
6030  FOR I = 1 TO 5
6040  VOLTS = VOLTS + C(I)
6050  CURRENT = CURRENT + C(I+5)
6060  NEXT
6070  AVVOLTS = VOLTS/5
6080  AVCURR = CURRENT/5
6090  KILL 1 TO 10
6100  PRINT AVVOLTS; "AVERAGE VOLTS"; 1; "VDC"
6110  PRINT AVCURR; "AVERAGE CURRENT"
```

When the results are output, the value for each individual channel will be suppressed, and the average value measured for volts and for amps will displayed in the form

```
S T 1 9:00:00.0
: AVERAGE VOLTS      P 001  1.51675  VAC  .
: AVERAGE CURRENT    P 002  0.75392  IAC  .
```

# Chapter 3
# Channel Definition


| *Para. No.* | *Title* | *Page No.* |
|---|---|---|
| 1 | Introduction | 3.2 |
| 1.1 | Channel Syntax | 3.2 |
| 2 | Analogue Input or 'Decimal' type Digital Input | 3.3 |
| 3 | Analogue Output | 3.5 |
| 4 | Status type digital Input | 3.6 |
| 5 | Event type digital Input | 3.8 |

# 1. INTRODUCTION

Before a channel or channel set may be used by a task it must first have been defined using the channel definition commands. This chapter details the channel commands associated with each of the input output channel types.

Analogue input
Decimal type digital input
Analogue output
Status type Digital input
Event type Digital input

## 1.1 CHANNEL SYNTAX

The following general rules apply when defining the channel setup.

i) The first entry in each new input string, when setting up, querying or changing channel data must always be the channel mnemonic **CH** and the channel number or channel set.

eg **CH** n-m <**SE** arg ><**CO**arg>C/R
**CH** n-m **CO**?C/R

ii) Channels may be specified in groups, however, all channels in a group must be of the same general type.

eg all strain gauges, all thermocouples, all digital outputs etc.

iii) All channels specified in any one channel definition must be of the same type:

ie either analogue inputs
or event/status input
or digital input/output

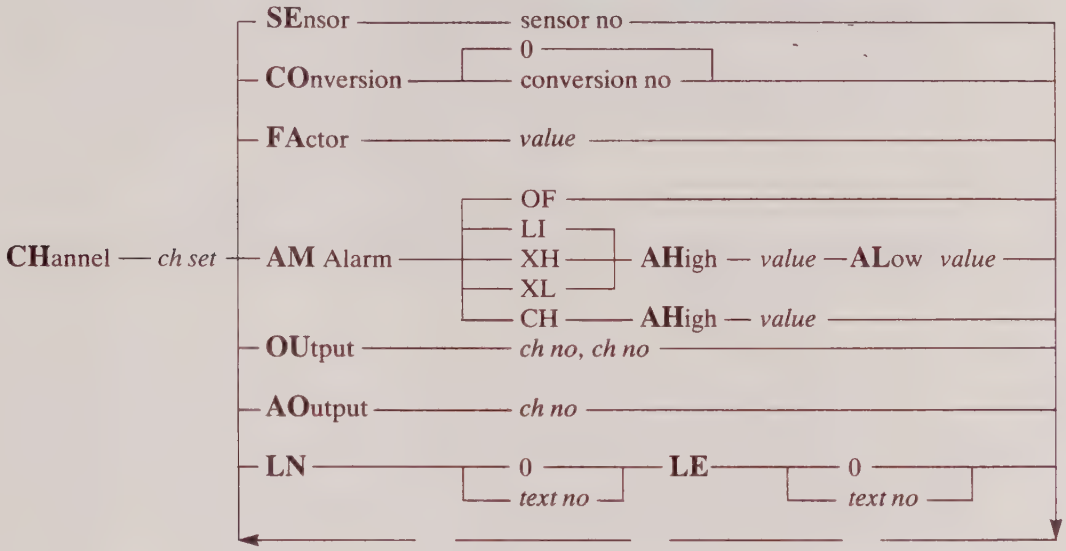
A mixture of types is not permitted.

iv) Most of the channel set up parameters are not essential commands. If not defined by the user they will assume their default values. However the channel mnemonic and channel set must be defined together with the sensor type for the channels.

eg **CH**n **SE**mmm C/R

## 2. ANALOGUE INPUT OR 'DECIMAL' TYPE DIGITAL INPUT

Command tree structure




The command tree shows the channel command associated with the definition of ORION Analogue input channels and decimal type digital input channels. The channel mnemonic, CH, and channel set must always be the first character entered in the command string, however, all other commands and related arguments are options and may be entered in any order, or omitted.

**CH** — *ch set*

**CH** The channel mnemonic.

*ch set* The channel numbers.

**SE**nsor — n

**SE** Select measurement type.

n The sensor number, a 3 digit number defining the measurement type, in the range:

110 to 115 voltage DC measurement
120 to 125 voltage AC measurement
210 to 279 Resistance measurement
310 to 365 Thermocouple measurement
410 to 479 Platinum Resistance thermomter measurement
510 to 514 Current DC
520 to 525 Current AC
600 to 685 Strain gauges
811 to 833 Digital input

Appendix 2, Sensor Number Table contains more detailed information on sensor numbers.

**CO**nversion —— n ——————

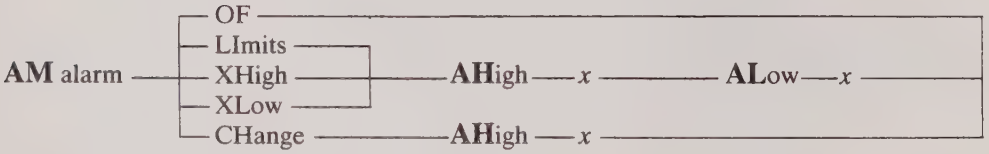
**CO** Define number of conversion routine which is to be applied to all logged data. The conversion will be applied to data for all tasks using the channel set. Default value is zero.

n 1-40: conversion routine number, or zero if no conversion to be applied.

**FA**ctor ——— *x* ——————

**FA** Define the gauge factor.

*x* the gauge factor.




**AM** Select the alarm type. (Default value OF).

| | |
|---|---|
| OF | Alarm off, but any parameters set up remain in memory |
| LI | Specify high and low limits, applicable to all SEnsors types except BIT |
| XH | Specify threshold value and lower limit of deadband applicable to all SEnsor types except BIT |
| XL | Specify threshold value and upper limit of deadband applicable to all SEnsor types except BIT |
| CH | Specify change size, applicable to all SEnsor types except BIT |

**AH** Specify upper limit of alarm or deadband, or change size.

*x* numeric value defining one of the following:
high alarm limit, threshold value, upper limit of deadband, change size.

**AL** Specify lower limit of alarm or deadband.

*x* numeric value defining one of the following:
low alarm limit, threshold value, lower limit of deadband.

**OU**tput Alarm ———— *n(lo), n(hi)* ——————

**OU** specify the two digital output channels associated with 'lo', 'hi', alarms.

*n* digital output channel number
On low alarm 'lo' goes from logic 'hi' to 'lo'
On high alarm 'hi' goes from logic 'hi' to 'lo'

**AO**utput —— *ch no* ——

**AO** specify the Analogue output to be associated with the input channel.

*ch no* Channel number of the Analogue Output Module channel that is to be 'attached' to the input channel.

**Note:** Each Analogue Output module has five output channels, the channel number will be in the range:

1 to 5 if module fitted in slot 4
21 to 25 if module fitted in slot 5
181 to 185 if module fitted in slot 13

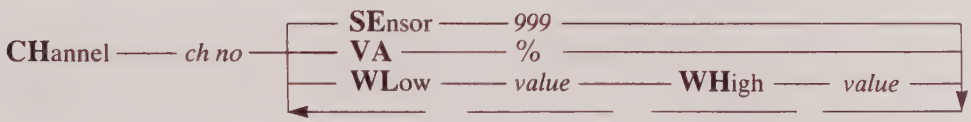
**LN** —— *n* —— **LE** —— *n* ——

**LN** Text to be output on normal result, specify text number. (0 for off).

**LE** Text to be output on alarm result specify text number (0 for off).

*n* 1-40: text number for text to be output with logged data or alarm data or events.

## 3. ANALOGUE OUTPUT




The command tree shows the channel commands associated with the definition or ORION analogue channels, i.e. the output channel associated with an ORION input channel.

The channel mnemonic, CH and output channel number must always be entered, however, the other commands and their arguments are optional and may be entered in any order or omitted.

**CH**annel —— *ch no* ——

**CH** Channel mnemonics.

*ch no* Channel number.

**SE**nsor —— *999* ——

**SE** select measurement type.

*999* sensor number for analogue output channel.

**VA**lue ——— *n* ———————

**VA** specify the set value (percentage) of the output required.

*n* percentage value (0 to 100%) to be used to obtain a fixed output in the range:
4 to 20mA if current is selected
0 to 10v if voltage is selected

where 0% = 4mA or 0V, 100% = 20mA to 10v.

**Note:** When a VAlue has been specified this will override any measured value obtained using the 'scale' facility.

**WL**ow ——— *n* ———————**WH**igh —— *n* ———————

**WL** define the minimum measurement value ie the value for 0% current or voltage output.

*n* the value for 0% full scale current or voltage output. (Note 1)

**WH** define the maximum measurement value, ie the value for 100% current or voltage output.

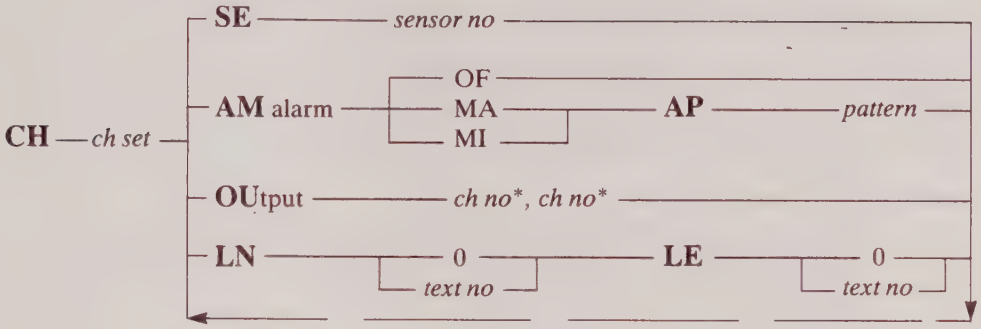
*n* the value for 100% full scale current or voltage output. (Note 1)

**Note 1:** The value, n, relates to the measured result for the associated input channel. Enter the numeric value only, engineering units, eg μ strain, °C, etc. will automatically be assumed depending on the type of sensor connected.

**Note 2:** Whenever the associated input channel is scanned or monitored the result obtained will cause the analogue output channel to produce a proportional current or voltage. This output level will be maintained until updated by the next scan or monitor measurement.

## 4. STATUS TYPE DIGITAL INPUT

Command tree Structure




*where the channel no. is repeated.

The command tree shows the channel commands associated with the definition of ORION Status type digital input channels. The channel mnemonic, CH, and the channel set must always be the first characters in the command string, however, all other command and related arguments are optional and may be entered in any order or omitted.

**CH**annel ——— *ch set* ———

**CH** the channel mnemonic.

*ch set* The channel numbers.

**SE**nsor ——— *n* ———

**SE** Select input measurement type.

*n* The sensor number, a 3 digit number, defining the measurement type, in the range:

| | |
|---|---|
| 701 to 708 | BIT type status input — Event status card |
| 711 to 715 | BCD Input — Event status card |
| 721 to 725 | Inverted BCD input — Event status card |
| 801 to 808 | BIT type status number — Digital I/O card |

Appendix 2, Sensor Number Table contains more detailed information on sensor numbers.

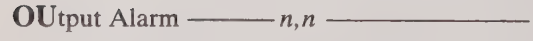
**AM** Select alarm type.

OF Alarm off but any parameters set up in memory remain.
MA Select alarm on pattern match, specify bit pattern with which input bit pattern should be compared for match. Applicable only to BIT type sensors.

MI Select alarm on pattern mismatch, specify bit pattern with which input bit pattern should be compared for mismatch. Applicable to BIT sensor type only.

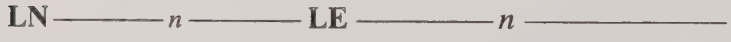
**AP** Set up bit pattern for alarm checking.

*x* 1-8 bits: Alarm pattern against which the input bit pattern will be compared. Channels which are to be ignored for alarm checking purposes should be masked by entering * in the relevant position in the pattern, the corresponding channel is output normally but disregarded by the alarm detector eg CH41 AP**010 where channels 41 to 45 have been defined as a channel group during sensor definition.




**OU** specify the digital output channel to be used for match/mismatch alarms.

*n* The digital output channel number. The channel number should be repeated separated by a comma.



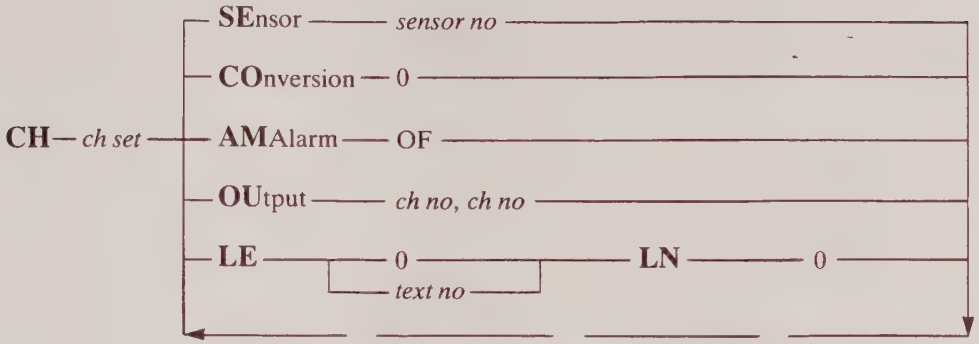

**LN** Text to be output on normal result, specify text number (default 0 for off).

**LE** Text to be output on alarm result specify text number (default 0 for off).

*n* 1-40: text number for text to be output with logged data or alarm data or events.

## 5. EVENT TYPE DIGITAL INPUT

Command tree structure




The command tree shows the channel commands associated with the definition of ORION Event type Digital input channels. The channel mnemonic CH, and channel set must always be the first characters in the command string, however, all other commands and related arguments are optional and may be entered in any order or omitted.

**CH**annel — *ch set*

**CH** the channel mnemonic.

*ch set* The channel numbers.

**SE** — *n*

**SE** Select input measurement type.

*n* The sensor number, a 3 digit number, defining the measurement type, in the range.

| | | |
|---|---|---|
| 740 to 742 | Totalise events | Event |
| 750 to 752 | Count events | Status |
| 760 to 762 | Event capture | module |
| 900 to 904 | Counter timer module – | Frequency |
| 910 | | Totalise |
| 920 to 925 | | Period |
| 930 | | Interval |
| 940 | Counter timer function off. | |

Appendix 2, Sensor Number Table, contains more detailed information on sensor numbers.

**CO**nversion ——— *n* ———————

**CO** Define conversion number as zero, ie no conversion applied.

*0* No conversion applied.

**AM** ——— OF ———————

**AM** Ensure that alarm monitoring is off.

OF No alarm monitoring.

**OU** ——— *ch no, ch no* ———————

**OU** Specify the digital output channel/s to be associated with the event task.

*ch no* the digital output channel number.

**LN** ——— *n* ——— **LE** ——— *n* ———————

**LN** Text to be output on normal result, specify text number (default 0 for off).

**LE** Text to be output on alarm result specify text number (default 0 for off).

*n* 1-40: text number for text to be output with logged data or alarm data or events.

# Chapter 4
# Task Definition


| *Para. No.* | *Title* | *Page No.* |
|---|---|---|
| 1 | Introduction | 4.2 |
| 1.1 | Task Syntax | 4.2 |
| 2 | Measurement task | 4.3 |
| 3 | Event task | 4.9 |

# 1. INTRODUCTION

The Tasks that can be set up using the ORION TASK commands are of two basic types:

**Measurement Scanning Tasks** used to scan analogue, status or counter timer inputs or event inputs that are to be counted or totalised.

**Event Tasks** used for detection of event inputs.

The sections that follow detail the task syntax and list the task commands associated with each of the tasks.

## 1.1 TASK SYNTAX

There are a number of general rules that can be applied when defining a task setup, these are detailed below:

i) When setting up, changing or querying task data the input string must start with the task mnemonics, **TA,** followed by the task number.

eg **TA** 1

ii) When entering the time as a parameter, in the form *dd-hh:mm:ss.t,* the tenths of the seconds need only be included in the argument if not zero. The value of the seconds should always be specified, but the minutes, hours and days need only be entered if they are significant.

eg enter 10.5 for 10½ seconds
1:00:00 for 1 hour (no minutes or seconds)

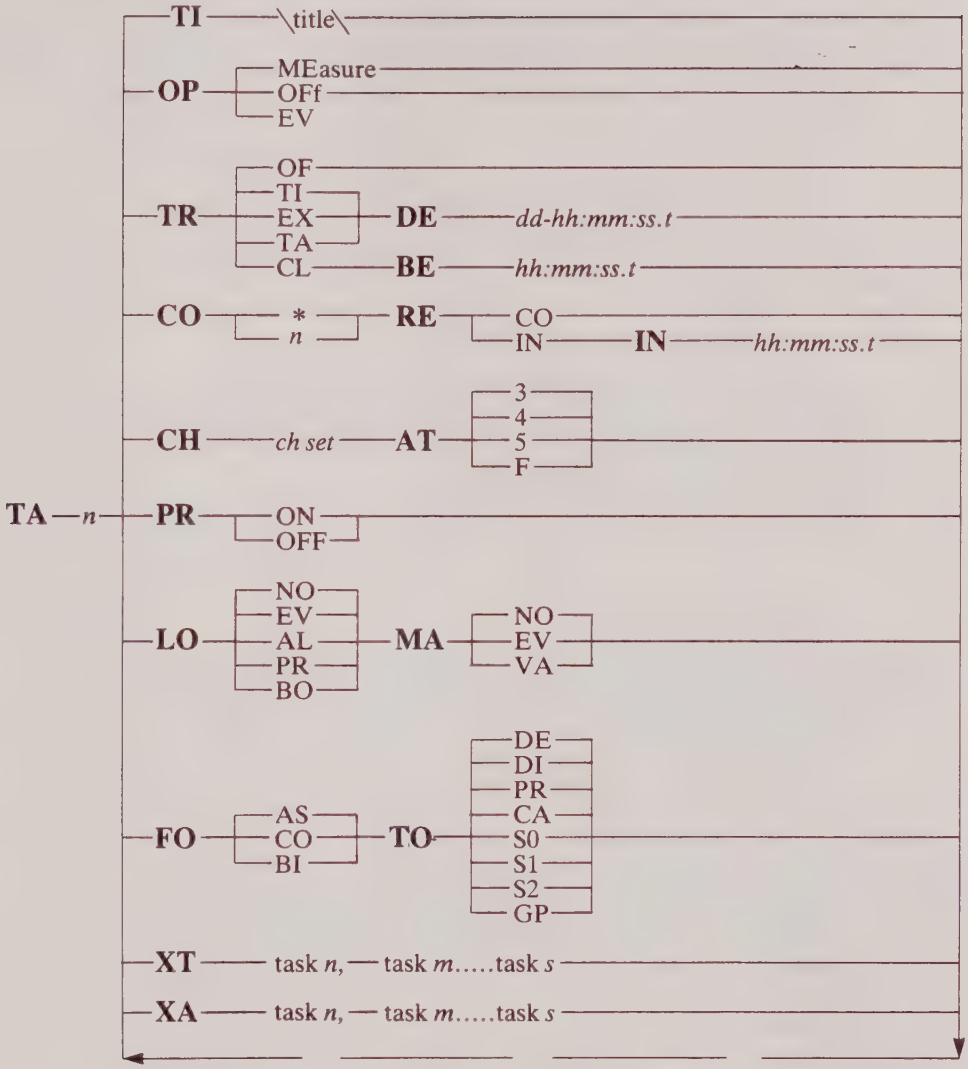
iii) When setting up a task all parameters except **TA**n, are optional. However, there are a number of parameters that must be entered before a task will run successfully these are:

Task number, **TA**n
Task type, Measurement **OP** ME or Event **OP** EV
Task trigger, **TR,** followed by relevant parameter defining trigger type
Channel set, **CH** ch set, the channel set must also be defined during channel definition.

All other parameters will assume their default values, if ORION delta has been initialised, or contain any values previously set-up if not.

## 2. MEASUREMENT TASK

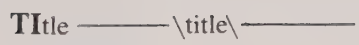
Command Tree structure




The command tree shows the commands associated with the definition of an ORION Scanning Measurement Task. The Task mnemonic, **TA,** and task number, *n* must always be entered, however all other commands are optional and may be entered in any order or omitted.

**TA** The task mnemonic.

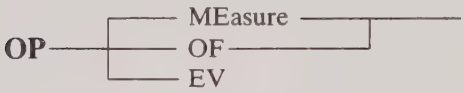
*n* The number of the task to be defined.




**TI** Title to be given to the task.

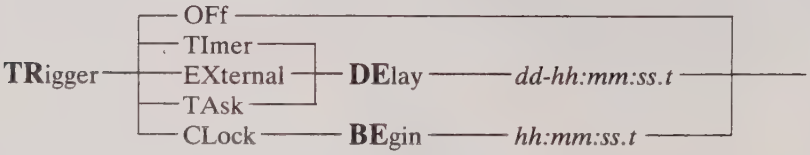
title 18 characters maximum: the title to be given to the task. The 'title' character string must be preceded by and followed by the delimiting character \ .

eg \START UP SEQUENCE\




**OP** Define task operation required. Default value OFf.

OF Will prevent a task from running if it has not yet been triggered. It will have no effect if entered when the task is running, but if the task is subsequently halted it will not restart when re-triggered.
ME Task n is a Scanning Measurement task.
EV The task n is an Event task (see subsection 3, for definition details.)




**TR** Task Trigger. Default value OFf.

OF Task will not respond to any trigger.
TI Trigger from internal timer, set delay time required.
EX Trigger externally from the execute command, set delay time required.
TA Trigger from another task, set delay time as required.
CL Trigger from internal 24 hour clock at the time specified by the begin command.

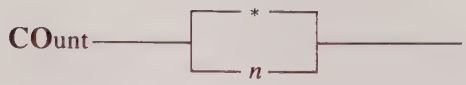
**Note:** the task will automatically retrigger at the same time 24 hours after the last begin time.

**DE** Specify delay before triggering task. Default value 00-00:00:00.0.

*dd-hh:mm:ss.t* Delay time required.

**BE** Specify start time. Default value 00:00:00.0.

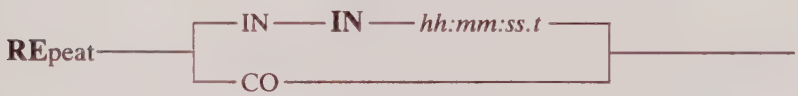
*hh:mm:ss.t* Start time for trigger from internal clock.




**CO** Define the number of scans required for the measurement/scanning task Default value * (continuous scanning).

* Continuous Scanning.

*n* The number of scans required. The maximum number that can made is 65535, numbers entered greater than this value will be treated as their modulo remainder, except for zero remainder which will be flagged as an error.




**RE** Define what the interval between scans is to be.
Default value is continuous, CO

IN There is to be a defined interval between scans. Interval triggering will initiate measurement scans at the defined interval, subject to task priority. If however the ORION buffer becomes completely filled, the system will wait for sufficient memory to become available before continuing with the scan.

CO Continuous triggering of measurement scans. When continuous triggering is selected the next scan will be triggered when the end of scan marker for the task has been passed to the output interface. This prevents the ORION buffer becoming completely filled as the system is slowed down to the speed of the interface.

**IN** Interval between measurement scans if scanning is not continuous, Default value is 00:00:00.0.

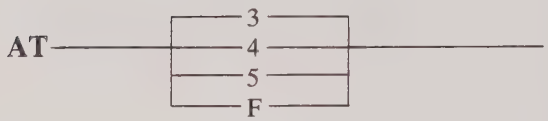
*hh:mm:ss.t* The measurement delay interval.

**CH**annel ——— *ch set* ———

**CH** Define the channel number/s or channel set to be included in the task.

*ch set* The channels may be specified either as a single channel number, several channel numbers separated by commas (,) or in the form Channel n to m.

eg TA1 CH5,7,10
TA2 CH11-22,4




**AT** Select the rate at which the measurements are to be made. Four different integration times are available these are set automatically according to the measurement rate selected. Default value is 3 (150 or 500 chan/sec).

| | |
|---|---|
| 3 | Measurement rate of 500 chan/sec if using Solid State Selector 150 chan/sec if using Reed Relay Selector |
| 4 | Measurement rate of 100 channels per second |
| 5 | Measurement rate of 40 channels per second |
| F | Measurement rate of 10 channels per second |

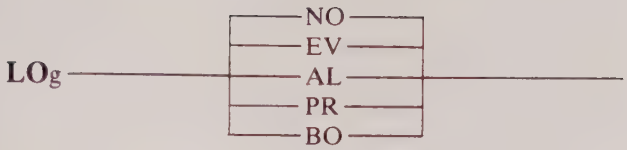
**Note:** The measurement rates are applicable to DC voltage based measurements, AC inputs are scanned at 5 chan/sec.

**PR** Select end of scan processing. This will extend the processing capabilities of the task as described in chapter 6.

When processing is selected the program associated with the task will be performed at the end of each scan. This program can be used to perform sophisticated data reduction on the measured channel data before it is output.

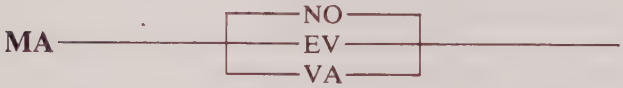
| | |
|---|---|
| ON | Perform end of scan processing before outputting result data. |
| OF | End of scan processing is not performed. |

**Note:** Although processing may be off, RUN and HALT markers, and Date change data may be sent to any output device(s) specified in the processing program, if such exists.



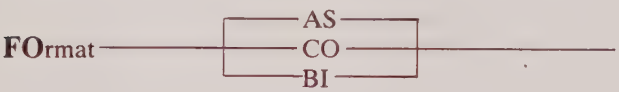

**LO** Define the type or types of data that are to be logged Default value is NOne.

| | |
|---|---|
| NO | No data to be logged |
| EV | Log all data |
| AL | Log only data for channels in alarm |
| PR | Log only processed data |
| BO | Log both alarm and processed data |




**MA** Specify when header markers are to be output. Default value: NOne.

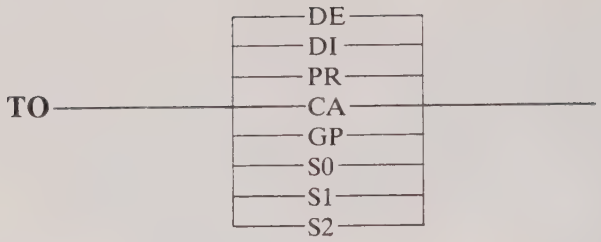
| | |
|---|---|
| NO | Header markers are not to be output |
| EV | Output a header each time the task is actioned |
| VA | Output a header only when valid data is logged |




**FO** Specify the format in which the ORION data is to be output to the specified interface. Default value is AS (full format).

| | |
|---|---|
| AS | Full format: 40ASCII characters. It includes text messages and titles in addition to full numeric information. |
| CO | Compact Format: 20 ASCII characters. It contains full numeric information, but excludes text. |
| BI | Coded format: 10ASCII characters. It contains results in binary format which has been encoded into printable ASCII characters. |

**Note:** If the result is a processed result it may be followed by an additional 10 character string.




**TO** Specify device to which data is to be logged. Default value is DE (null device).

| | |
|---|---|
| DE | Null device, data is lost |
| DI | Front panel or local terminal |
| PR | Front panel printer, 3530D only |
| CA | Front panel magnetic cartridge, 3530D only |
| GP | GPIB interface socket on the GPIB board 35304A |
| S0 | The standard RS232 interface on the controller board |
| S1 | Socket PL1 on the serial interface board 35304B |
| S2 | Socket PL2 on the serial interface board 35304B |

**XT**rigger _____ task *n*, task *m*, .... task *t*_____

**XT** To specify which task or tasks are to be cross triggered on an alarm condition.

*n..t* The number or numbers of the tasks that are to be triggered up to a maximum of eight tasks, 1 to 8.

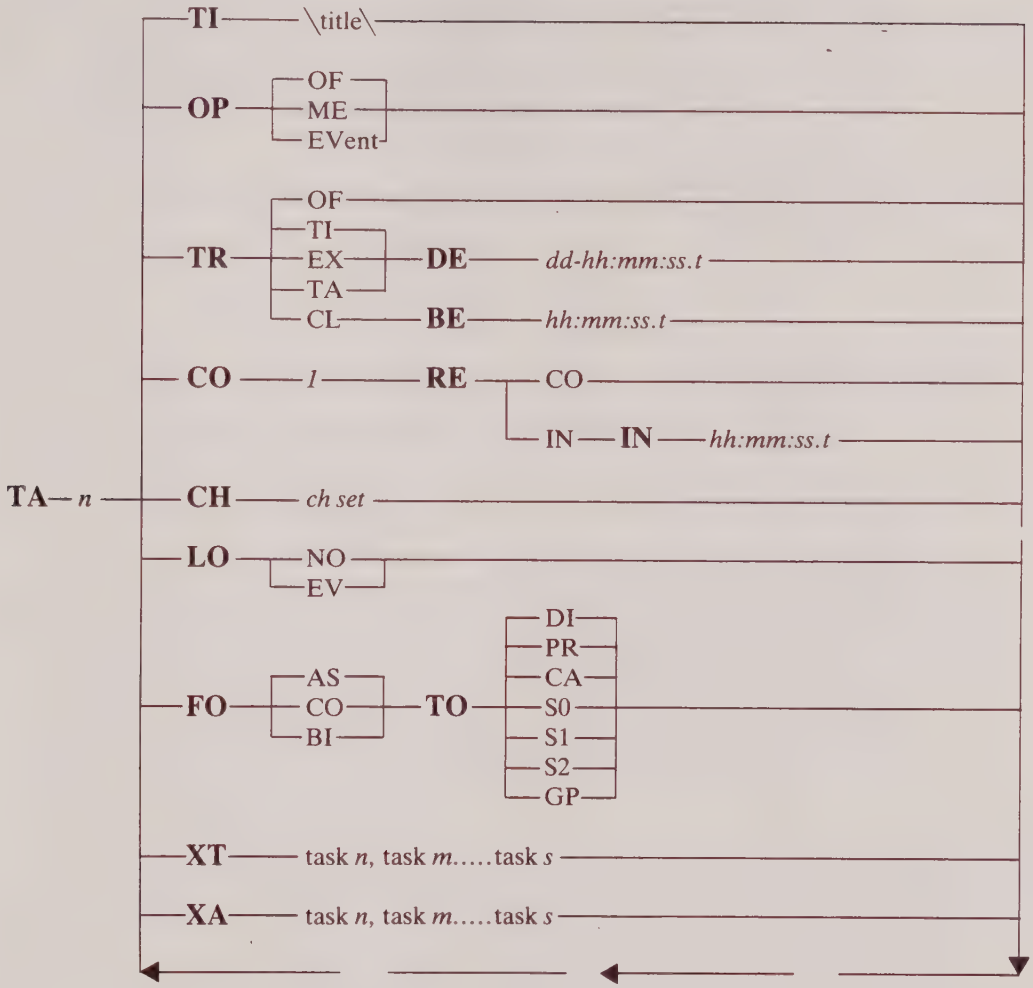
**XA**bort _____ *task n*, task *m*, .... task *t* _____

**XA** Specify the number of tasks or task that are to be aborted on alarm condition.

*n..t* The number or numbers of tasks that are to be aborted on an alarm condition.

## 3. EVENT TASK

Command Tree Structure




The command tree shows the commands associated with the definition of an ORION Event task. The task mnemonic, TA, and task number, n, must always be entered, however all other commands are options and may be entered in any order or omitted.

**TA**sk ——— *n* ———

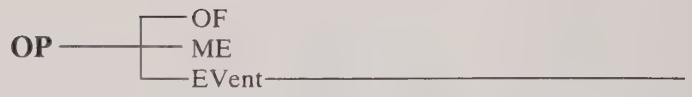
**TA** The task mnemonic.

The number of the task to be defined

**TI**tle ——— \title\ ———

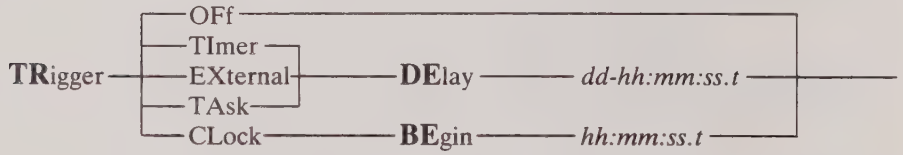
**TI** Title to be given to the task.

title 18 characters maximum: the title to be given to the task. The 'title' character string must be preceeded by and followed by the deliminating character\.




**OP** Define task operation required. Default value is OFf.

OF Will prevent a task from running if it is has not yet been triggered. It will have no effect if entered when the task is running, but if the task is subsequently halted it will not restart when re-triggered.
EV Task n is an Event task.
ME The task is a scanning measurement task, see subsection 2 for definition details.




**TR** Task Trigger Default value OFf.

OF Task will not respond to any trigger.
TI Trigger from internal timer, set delay time required.
EX Trigger externally from the execute command, set delay time required.
TA Trigger from another task, set delay time as required.
CL Trigger from internal 24 hour clock at the same time specified by the begin command.
**Note:** the task will automatically retrigger at the same time 24 hours after the last begin time.

**DE** Specify delay before triggering task Default value is 00-00:00:00.0.

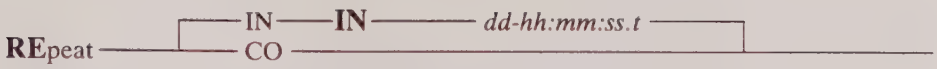
*dd-hh:mm:ss.t* Delay time required.

**BE** Specify start time. Default value 00:00:00.0.

*hh:mm:ss.t* Start time for trigger from internal clock.

**CO** Define the number of scans required for an event task, this should always be defined as 1. The default value is *, continuous.

*1* For an event task the scan count should always be 1.




**RE** Define whether event logging is to be continuous or of a fixed duration.

CO Continuous monitoring of events.
IN The event task is to be of a fixed duration.

**IN** Define the duration of the event task. Default value is 00-00:00:00.0

*dd-hh:mm:ss.t* The task duration.

**CH** Define the channel number of numbers to be included in the event task.

*chan set* The channel may be specified singularly or a channel set.
eg TA1 CH5,7,10,4
TA2 CH11-22,24-26

**LO** Define the data to be logged. Default value is NO.

NO No data to be logged
EV Log all data

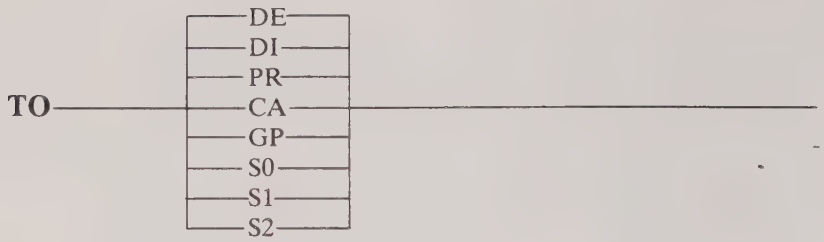
**FO** Specify the format in which the ORION data is to be output to the specified interface.

AS Full format: 40ASCII characters; It includes text messages and titles in addition to full numeric information.

CO Compact Format: 20 ASCII characters; contains full numeric information, but excludes text.

BI Coded format: 10ASCII characters; containing results in binary format which has been encoded into printable ASCII characters.

**Note:** If the result is a processed result it may be followed by an additional 10 character string.




**TO** Specify device to which data is to be loggd. Default value is DE (null device).

| | |
|---|---|
| DE | Null device, data is lost |
| DI | Front panel or local terminal |
| PR | Front panel printer, 3530D only |
| CA | Front panel magnetic cartridge, 3530D only |
| GP | GPIB interface socket on the GPIB board 35304A |
| S0 | The standard RS232 interface on the controller board |
| S1 | Socket PL1 on the serial interface board 35304B |
| S2 | Socket PL2 on the serial interface board 35304B |

**XT**rigger _____ task *n*, task *m*, .... task *t* _____

**XT** To specify which task or tasks are to be cross triggered on an alarm condition.

*n..t* The number or numbers of the tasks that are to be triggered up to a maximum of eight tasks 1 to 8.

**XA**bort _____ task *n*, task *m*, .... task *t* _____

**XA** Specify the task or tasks that are to be aborted on alarm condition.

*n..t* The number or numbers of the tasks that are to be aborted on an alarm condition, in range 1 to 8.

# Chapter 5
# Control and other Commands


| *Para. No.* | *Title* | *Page No.* |
|---|---|---|
| 1 | Control Commands | 5.2 |
| 1.1 | RUn | 5.2 |
| 1.2 | HAlt | 5.2 |
| 1.3 | PAuse | 5.2 |
| 1.4 | GO | 5.2 |
| 1.5 | Cross trigger | 5.2 |
| 1.6 | Cross Abort | 5.2 |
| 2 | Initialisation command | 5.3 |
| 3 | Cartridge command | 5.4 |
| 4 | Monitor command | 5.6 |
| 5 | Measure command | 5.7 |
| 6 | List command | 5.8 |
| 7 | Renumber command | 5.9 |
| 8 | Drift command | 5.9 |
| 9 | Set command | 5.9 |
| 10 | Write message command | 5.10 |
| 11 | Conversion Definition command | 5.11 |
| 12 | Text Definition command | 5.12 |
| 13 | Query command | 5.12 |
| 13.1 | Querying a parameter | 5.12 |
| 13.2 | Query only commands | 5.13 |

# 1. CONTROL COMMANDS

## 1.1 RUN

**RU** Start tasks: all programmed tasks become operational. The ORION logger awaits the first valid task trigger, then triggers each task when appropriate.

When the RUn command has been given, ORION tasks may be queried using the query command, see section 13, but tasks may not be modified, and no further tasks may be defined.

## 1.2 HALT

**HA** Stop task: all tasks running or awaiting a valid trigger are halted. Task and channel setup data may be entered or modified whilst the ORION delta is in the halted state.

## 1.3 PAUSE

**PA** Pause: does not HAlt the operation of the tasks running but suppresses the output of data to the interfaces. The programmed tasks will continue to run but data will be buffered up until the GO command is used to cancel the PAuse if however the buffer becomes full the tasks will be temporarily suspended until data has been output and there is room in the buffer for further data.

## 1.4 GO

**GO** Continue: cancels the pause. All programmed tasks are fully operational, buffered data is output to the relevant device.

## 1.5 CROSS TRIGGER

**XT**rigger n,m,....p

**XT** Cross Trigger: trigger specified task or tasks.

*n,m....p* The number of the task or tasks to be triggered. Up to eight tasks may be included in the command, each separated by a comma (,).

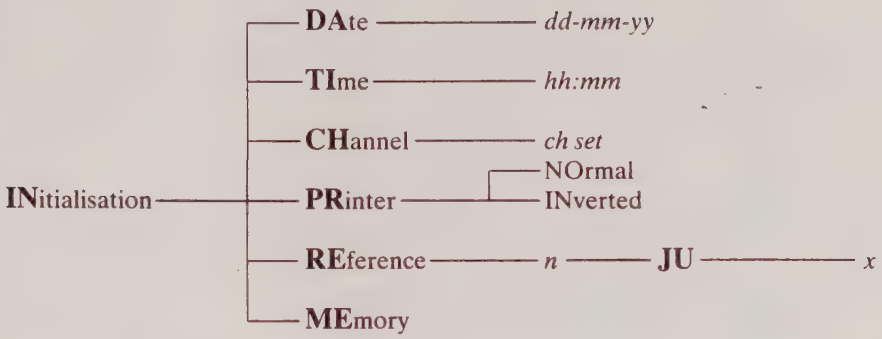
## 1.6 CROSS ABORT

**XA**bort n,m,....p

**XA** Cross Abort: abort the specified task or tasks.

*n,m....p* The number of task or tasks to be aborted. Up to eight tasks may be included in the command, each separated by a comma (,).

## 2. INITIALISATION COMMAND




**IN** Initialise date, time, printer etc.

**DA** Set up current data in internal clock.

*dd-mm-yy* current date.

**TI** Set up current time held in the internal clock.

*hh : mm* time.

**CH** Initialise analogue channels specified: this measures and stores 'unstrained' parameters from strain gauge channels and sets up auto ranging.

*ch set* channel numbers to be initialised.

**PR** Initialise printer (3530D only) to print in normal or inverted mode.

NO Normal = top reading is most recent data.
IN Inverted = bottom reading is most recent data.

**RE** Select external thermocouple reference junctions.

*n* 1 to 4: number of reference junction.

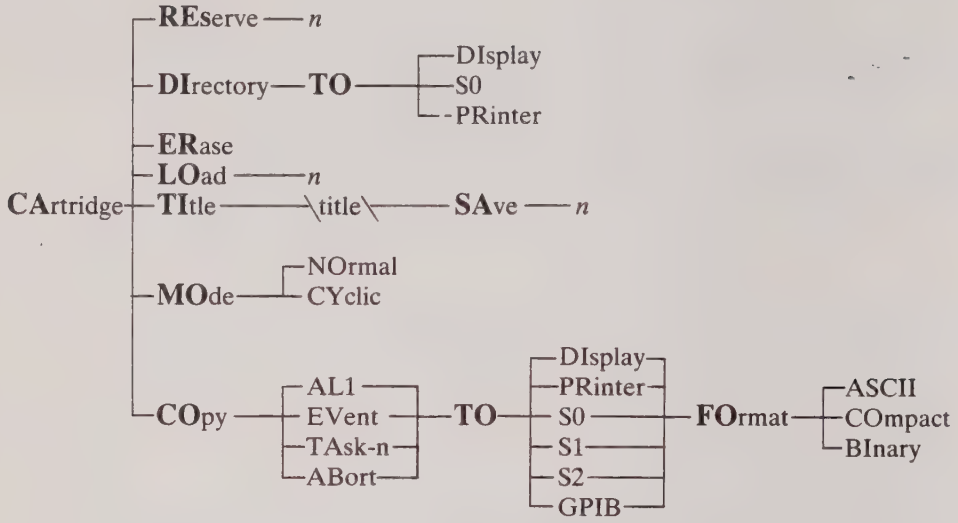
**JU** External reference temperature.

*x* Junction temperature in °C.

**ME** Reset memory to initial switch-on state; the data base values will return to their default values.

**Note:** any remote device will lose command.

## 3. CARTRIDGE COMMAND


REserve — n
DIrectory — TO — DIsplay / S0 / PRinter
ERase
LOad — n
CArtridge — TItle — \title\ — SAve — n
MOde — NOrmal / CYclic
COpy — AL1 / EVent / TAsk-n / ABort — TO — DIsplay / PRinter / S0 / S1 / S2 / GPIB — FOrmat — ASCII / COmpact / BInary

**CA** Tape commands: Applicable to 3530D only. Use of **CA** command with ORION F will result in an error prompt.

**RE** reserve space for programs on tape.

*n* 1 to 7: number of program spaces to be reserved.

**DI** List program numbers and names on device specified.

**TO** specify device to which directory is to be output.

DI front panel display.
S0 standard RS232 on controller board.
PR front panel printer.

**ER** Erase logged data but not the programs.

**LO** load the specified program from cartridge into memory.

*n* 1 to 7: program to be loaded into memory.

**TI** Specify title to be given to program to be saved.

\title\ up to 40 character title. The delimiting character \ must be used before and after the character string, eg \TEST PROG.A\.

**SA** Save program on cartridge. For strain gauge channels any initialised value will be saved.

*n* 1 to 7: the program in memory will be saved with this reference number.

**MO** specify recording mode.

NO normal mode: write data to tape until cartridge is full.
CY cyclic mode: write data to tape until cartridge is full, then rewind to start of data and overwrite with new data.

**CO** cause the data specified to be copied from the cartridge to an external output device.

AL all data
EV event data, applicable only to data recorded in binary coded format.
TAn the data from the single task no. specified, applicable only to data recorded in binary coded format.
AB stop copying data from cartridge.

**TO** specify device to which data from cartridge is to be copied.

DI front panel display or local terminal.
PR front panel printer.
S0 standard RS232 on controller board.
S1 socket PL1 on serial interface board 35304B.
S2 socket PL2 on serial interface board 35304B.
GP GPIB interface socket on GPIB board 35304A.

**FO** Select format in which data is to be output.

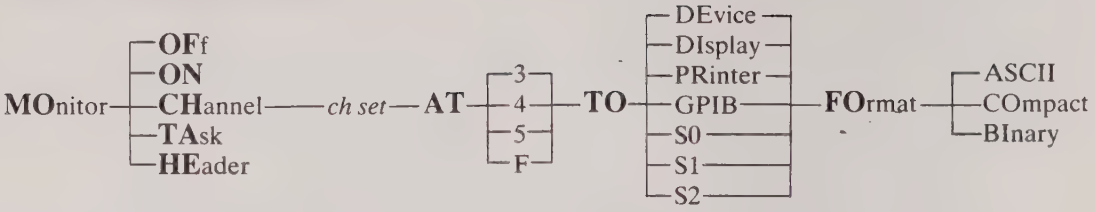
AS Full format: 40 ASCII characters per channel, including text.
CO Compact format: 20 ASCII characters per channel, without text.
BI Binary encoded: 10/20 ASCII characters per channel depending on whether the data is processed data.

The format which may be output is conditional upon the format in which the record was made:
**Record full format:** Copy in full format only.
**Record compact format:** Copy compact format only.
**Record binary encoded:** Copy output may be any one of the three selections:

1. Binary encoded: the output device must be able to use (understand) the code, or, if for display or hard copy, must decode to decimal format.
2. Compact format: decoded within ORION.
3. Full format: decoded within ORION.

**Note:** When decoding binary coded data for output in either full or compact format the ORION must contain the program under which the data was recorded. The program must be identical in every respect to the original program particularly if the tape contains processed data otherwise the data will not be decoded correctly.

## 4. MONITOR COMMAND




**MO** Set-up the parameters for the monitoring facility; initiate or stop the monitoring of channels, tasks or headers.

**Note 1:** The monitor command can be set up and initiated when a task is running.

**Note 2:** Monitoring of the specified channels, task or header will start immediately that command input string has been entered.

**OF** Monitoring stops but all parameters remain in set-up state.

**ON** Select ON when monitoring parameters have been set-up to start monitoring of channels, tasks or headers.

**CH** Select monitoring of channels, when the monitor scan is initiated results are output to device specified by **TO** command.

*ch set* channel numbers to be monitored.

**AT** Define measurement rate required Default value is 3

| | |
|---|---|
| 3 | measure at 500 or 150 channels per second |
| 4 | measure at 100 channels per second |
| 5 | measure at 40 channels per second |
| F | measure at 10 channels per second |

**TO** Specify device to which data is to be output. Default value DE.

| | |
|---|---|
| DE | Null device, output lost. |
| DI | Front panel display or local terminal |
| PR | Front panel printer (3530D only) |
| GP | GPIB interface socket on GPIB board 35304A |
| S0 | Standard RS232 socket on controller board |
| S1 | Socket PLI on serial interface board 35304B |
| S2 | Socket PL2 on serial interface 35304B |

**Note:** Although it is possible to specify output to cartridge (CA) this will cause a software interrupt and should not be used.

**FO** Format in which data is to be output. (Default value ASCII).

| | |
|---|---|
| AS | Full format 40 ASCII characters; text is included. |
| CO | Compact 20 ASCII characters; no text included. |
| BI | Binary encoded 10 ASCII characters data values are converted to printable ASCII characters. |

**TA** Select task monitoring, results are output to device specified by **TO** command when the monitored task is running.

*n* specify task to be monitored.

**HE** Select monitoring of headers, start task marker (STn) and time of start and End Task marker (DTn) are output at the beginning of each scan respectively.

## 5. MEASURE COMMAND

**ME** Immediate measurement: this command will cause an immediate measurement to be made on the single channel specified. The response to the measurement is sent to the device which issued the command. The integration will be ignored when measuring on digital channels. A short form of the command (**ME**n) may be used in which case the **CH** mnemonic is not required and the integration and format values set up for a previous measure are used.

**AT** Specify measurement integration. Default value 3.

| | |
|---|---|
| 3 | Measurement rate 500 or 150 channels per second |
| 4 | Measurement rate 100 channels per second |
| 5 | Measurement rate 40 channels per second |
| F | Measurement rate 10 channels per second |

**FO** Format in which result is to be output. Default value AS.

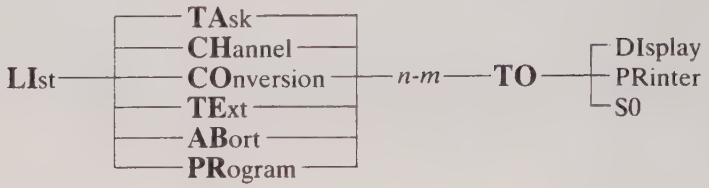
| | |
|---|---|
| AS | Full format 40 ASCII characters; text is included. |
| CO | Compact format 20 ASCII characters; text is not included. |
| BI | Binary encoded 10 ASCII characters; data values are converted to printable ASCII. |

**CH** Specify number to be measured.

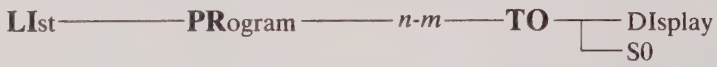
*n* Channel number to be measured.

## 6. LIST COMMAND

ORION Delta (3530D) only




for an ORION F (3530F) only




**LI** List the existing set up parameters of the required function on the device specified. ORION F is restricted to the listing program lines only on either the Standard RS232 socket on the controller board or the front panel socket.

**TA** List the current set-up parameters for the task or tasks specified.

**CH** List the current set-up parameters for all channels present in the range specified.

**CO** List the current conversion set-up for conversion numbers specified.

**TE** List the text specified by the text number/s.

**PR** List the program or programs specified.

**AB** Abort the current list command.

*n-m* The value of n and m are dependent on the function specified.
Task: value in range 1 to 8 a single task may be specified or a group of Tasks, eg 1 or 1-5.
Channel: value in range 1 to 200, a single channel or group of channels may be specified. Only details of channels actually present will be listed.
Conversion number: value in range 1 to 40, specified singularly as a group, eg 3 or 10-15.
Text: value in range 1 to 40, specified singularly or as a group.
Program: line numbers for program to be listed, eg 1000-1100.

**TO** Specify device to which output is to be sent.

DI Front panel display. (RS232)
PR Front panel printer. (3530D only)
S0 Standard RS232 socket on controller board.

## 7. RENUMBER COMMAND

**RE**number —— *Tnnn, Tmmm, pp,*

**RE** Renumber program lines.

*Tnnn* Starting at program line number nnn of task T (Old line number).

*Tmmm* Beginning with the line number of Tmmm (New line number).

*pp* In increments of pp.

## 8. DRIFT COMMAND

**DR** Specify whether drift compensation is required.

ON Periodic measurements of zero are made for each range at appropriate resolutions and the result stored.
These readings are used as corrections when the active channels are scanned.

OF No drift compensation to be applied.

## 9. SET COMMAND

**SE** Set the required bit pattern in a channel set, or the percentage value for an Analogue output Channel.

*ch set* The channel number or channel set.

**TO** Specify the bit pattern required.

*n* Bit pattern (binary value).

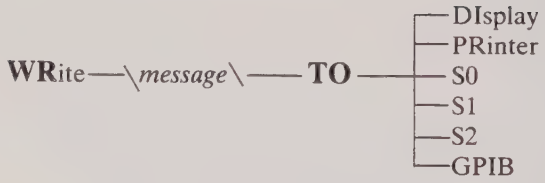
**VA** Specify the percentage value to be used for the output value.

*m* Percentage value to be used to obtain a fixed output in the range:

4 to 20mA if current is selected
0 to 10v if voltage is selected

where: 0% = 4mA 0v, 100% = 20mA or 10v.

## 10. WRITE MESSAGE COMMAND



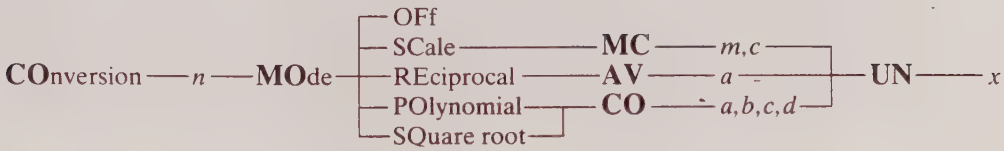

**WR** Send a message

*message* 40 characters max. message to be sent to device specified. The delimiting character \ must be used before and after the character string eg \I HAVE TAKEN CONTROL\ .

**TO** Specify device to which message is to be sent.

| | |
|---|---|
| DI | Front panel display or terminal |
| PR | Front panel printer |
| S0 | Standard RS232 socket on controller board |
| S1 | Socket PL1 on serial interface board 35304B |
| S2 | Socket PL2 on serial interface board 35304B |
| GP | GPIB interface socket on GPIB interface board 35304A |

## 11. CONVERSION DEFINITION COMMAND




**CO** Conversion factor set up.

*n* 1 to 40: conversion routine number, default value 0.

**MO** Specify convversion mode to be used.

| | |
|---|---|
| OF | Deselect the numbered conversion routine but leave parameters in memory. |
| SC | Scaling and offset. (mx+c,) specify values using MC command. |
| RE | Reciprocal, (a/x) specify value of 'a' using AV command Polynomial. |
| PO | Polynomial ($Ax^3 + bx^2 + cx+d$), specify values using CO command |
| SQ | Square root, $((\sqrt{(ax+b)})*cd)$, specify values using CO command. |

**MC** Specify scaling values 'm' and 'c'.

*m,c* Scaling 'm' and offset 'c', decimal values to be used.

**AV** Specify value of 'a'.

*a* Decimal value.

**CO** Specify values of a, b, c, d for polynomial, or square root.

*a,b,c,d,* Decimal values.

**UN** Units of result.

*x* Three characters specifying unit of results eg VDC.

## 12. TEXT DEFINITION COMMAND

**LA**bel——*n*——**TX**—— text

**LA** Define text to be output with normal or alarm data. Text is applied to the channels specified by the channel definition commands.

*n* 1 to 40: text number.

**TX** Define text string.

text 18 characters max. text string.

## 13. QUERY COMMAND

The function of the query command is to interrogate the ORION data base. Queries do not change the state of the data base and therefore unlike commands, may be made at any time ie by a device not in full 'LOCAL' or 'REMOTE' control.

All parameters may be queried with the following exceptions:

(i) The first parameter in an input command string or single command string:
eg **TA** task, **PA** pause, **DR** drift compensation
**CO** conversion, **MO** monitor, **CH** channel etc.
(ii) The printer mode, ie whether it is operating in normal or inverted mode.
(iii) Tape mode, ie whether storage is in normal mode or cyclic mode.

Any of the above will result in the error response E003, ie no query available.

### 13.1 QUERYING A PARAMETER

To query a parameter value the command mnemonic must be entered followed by a question mark (?) which replaces the normal parameter mnemonic or value.

eg **TA**1 **TR**? will query the task trigger defined at present for task 1.

The current value for the queried parameter, held in the ORION data base, is displayed at the terminal from which the enquiry was made.

For the example above one of the values: OF, TI, EX, IN or CL would be displayed.

**Note:** The reply to a query will always be returned in the format in which the parameters would normally be entered.

eg A query for the current date held in ORION **IN DA**?
response would be in the form *dd-mm-yy*

## 13.2 QUERY ONLY COMMANDS

Three commands are available which are solely used to query the interface cards present in the ORION. Their format and functions are as follows:

(i) Query interface type: **TY**

**CH**——*ch set*——**TY**?

The ORION replies with one of the following:

DI For a digital I/O card in the slot for corresponding channels.

AN An ANALOGUE card either reed relay, solid state or Analogue Output in slot for corresponding channels.

CO Counter timer module in slot for corresponding channels.

DA Digital to Analogue converter module in slot for corresponding channels.

EV EVENT/STATUS module in slot for corresponding channels.

E254 There is no card present in slot or there is a channel conflict, ie channels specified are on two different cards.

(ii) Query digital status: **DS**

**CH**——*ch set*——**DS**?

The ORION replies with either

DI For digital alarm mode, where an alarm is checked for a bit pattern match or missmatch.

AN In all other cases.

(iii) Query Analogue Output module configuration. The voltage or current drive capability is switch selectable on the Analogue Output card, but the status of a given channel may be queried by: **VI**

**CH**——*ch no*——**VI**?

The ORION replies with either:

VO Configured for voltage output.

CU Configured for current output.

# Chapter 6
# The Orion Processing Program


| *Para. No.* | *Title* | *Page No.* |
|---|---|---|
| 1 | Introduction | 6.2 |
| 2 | Processing Facilities | 6.2 |
| 3 | The Program Structure | 6.3 |
| 3.1 | Line numbers | 6.3 |
| 3.2 | Statements | 6.3 |
| 3.3 | Statement Separators (:) | 6.4 |
| 4 | Creating a program | 6.5 |
| 4.1 | Entering a program | 6.5 |
| 4.2 | Correcting a line | 6.5 |
| 4.3 | Deleting a line | 6.5 |
| 4.4 | Inserting a line | 6.5 |
| 4.5 | Renumbering program lines | 6.6 |
| 4.6 | List a program | 6.6 |
| 5 | Program Constants | 6.7 |
| 6 | Program variables | 6.7 |
| 6.1 | Predefined variables | 6.7 |
| 6.2 | User defined variables | 6.10 |
| 7 | Mathematical Operators and Functions | 6.10 |
| 8 | Control Structures | 6.11 |
| 8.1 | Decision making – IF THEN ELSE ENDIF | 6.11 |
| 8.2 | Program loops – FOR . . . NEXT | 6.12 |
| 8.3 | Unconditional branch – GOTO | 6.13 |
| 9 | The Output Statements | 6.14 |
| 9.1 | The PRINT Statement | 6.14 |
| 9.2 | PRINT Statement Extensions | 6.16 |
| 9.3 | Output Format Declaration | 6.17 |
| 9.4 | The WRITE Statement | 6.18 |
| 9.5 | The KILL Statement | 6.20 |
| 9.6 | SET . . . TO Statements | 6.20 |
| 10 | Digital Inputs | 6.21 |
| 11 | Control of Other Tasks | 6.22 |
| 11.1 | Triggering Other Tasks | 6.22 |
| 11.2 | Aborting Other Tasks | 6.22 |
| 12 | Counters and Timers | 6.22 |
| 12.1 | Counters COUNT1 to COUNT8 | 6.22 |
| 12.2 | Timers TIMER0 to TIMER9 | 6.22 |
| 13 | Random Channel Access | 6.23 |
| 14 | Memory Capacity | 6.24 |
| 14.1 | Processing Program Memory Requirements | 6.24 |
| 14.2 | Statement Memory Allocation | 6.25 |
| 15 | Speed of Execution | 6.26 |
| 16 | Programming Techniques | 6.26 |
| 17 | Vocabulary of Functions | 6.28 |

# 1. INTRODUCTION

The ORION processing program provides the ORION delta or ORION F user with a data manipulation facility which can be incorporated into a task to act on measured data before it is logged to an output device.

An ORION processing program is written using a high level language and may be used to perform both simple and sophisticated manipulation of channel data. No previous programming experience is required to write a program, however, users familiar with the BASIC programming language will recognise certain similarities in some of the ORION statements.

The ORION delta and ORION F can be programmed remotely via the interface ports using simple high level statements. Chapters 8 to 11 give details of the protocol requirements of each remote interface.

If using an ORION delta the program may optionally be saved on cartridge, and subsequently reloaded using the integral cartridge unit, Chapter 7 gives details of ORION 'Tape programs'.

For a processing program to be performed it must have been selected during task definition, see Chapter 4 section 2. A processing program will only act on channel data associated with a task that has been defined during Channel and task definition.

This chapter contains information required to program the ORION from a remote terminal and includes a brief description of the language elements. Chapter 12 of the 3530 ORION delta Operating Manual contains more detailed information of the programming language and programming techniques and should be used in conjunction with this manual.

# 2. PROCESSING FACILITIES

The following is a summary of the data processing facilities available using a processing program:

a) Individual channel results can be modified by the various mathematical operations, such as scaling, conversions involving logarithms and trigonometric functions etc. These operations can augment, or replace, the conversion facilties available under "Channel Definition".
b) Cross-channel calculations, such as ratios, power measurements (V × I) etc.
c) Calculations involving time, such as averages, statistical behaviour, control algorithms etc.
d) "PRINT" statements to direct selected results to DISPLAY, PRINTER and peripheral interfaces.
e) Making recorded results more meaningful by adding explanatory text to them.
f) The cross-triggering of other tasks under special circumstances.
g) The cross-aborting of other tasks under special circumstances.
h) The control of analogue and digital outputs.
i) Decision Processing. The logger can test results such that, IF a defined condition is found, THEN a prescribed action is taken, ELSE an alternative action is implemented. This decision making may include combinations of conditions that involve AND and OR. It can be applied to analogue and digital results.

# 3. THE PROGRAM STRUCTURE

A processing program will consist of consecutively numbered statements or lines in the form:

| | line number | statement |
|---|---|---|
| eg | 1001 | POWER = C1 * C2 |
| | 1002 | PRINT POWER |

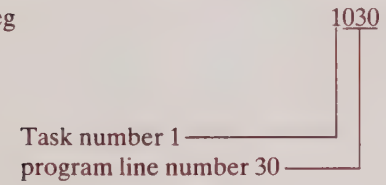
When a program is executed the statements will be obeyed in the order of their line numbers, unless the program statements indicate to the contrary.

## 3.1 LINE NUMBERS

The first item of each program line must be a four digit line number in the form Tnnn where:

T = first digit, indicating the number of the tasks of which the program forms part.
nnn = remaining three digits, identifying the line as one of a series of program lines.

eg




Valid line numbers are in the range 1000 to 8999.

A line number serves to identify the input to the logger as part of a processing program and not an ORION command language command (see chapter 3). It also identifies the task with which the program is to be associated. In addition the line number may be used by the program as an address or label for movements within that program.

eg 2010 GOTO 2080

will cause the program to jump from line 2010 to line 2080 ignoring any statements that are contained in intervening lines.

## 3.2 STATEMENTS

A line of the program will consist of one of more statements, each statement uses common English words and familiar mathematical notations. In general a statement will consist of one or more 'control words' and an 'expression'.

eg 1050 IF C1 <> 10 THEN: GOTO 1100:ENDIF

The above example contains three statements:

| 1050 | IF C1 <> 10 THEN | GOTO 1100: | ENDIF |
|---|---|---|---|
| line number | statement 1 | statement 2 | statement 3 |

When the amount of memory available for programs is not a problem the user may prefer to assign a new line number to each statement. This technique has three advantages:

1. Improves the clarity of the program.

2. Avoids the risk of exceeding the line capacity, ie 71 characters. (Note 1).

3. Makes program modification easier.

The example would then read:

```
1050    IF C1 <> 10 THEN
1051    GOTO 1100
1052    ENDIF
```

When writing large programs it may be necessary to use multi-statement lines in view of the fact that each program line will always occupy 88 bytes of memory, regardless of the number of statements included in that line, see Note 1.

**Note 1:** The maximum number of characters that can be input in a program line is theoretically 71. However, in practice the actual number of characters that a line can contain is considerably less and will be dependent on the number and type of statements included in that line. Every program line is allocated a fixed number of bytes of program memory, ie 88 bytes, of which each statement will occupy a variable number depending on content. For further information refer to section 14.2.

### 3.2.1 Control words

The statement control words are of four types:

a) Program control, eg IF . . . THEN, GOTO STEP.
b) Data Output control, eg PRINT, WRITE,KILL
c) Output control, SET TO
d) Task control, eg TRIGGER, ABORT

### 3.2.2 Expression

The expression component of a statement may consist of one or a number of the following component parts.

a) Variable-simple or indexed, eg MAX, TOTAL (I), described in section 6.
b) Constant-numeric, exponential, textural, eg 1.5, 4.375E−5,"OVERLOAD", described in section 5.
c) Operator-mathematical, relational, eg +,−,>, as described in section 7.
d) Function-Mathematical functions, eg ABS(x), SIN (x), as described in section 7.

## 3.3 STATEMENT SEPARATORS (:)

Where multi statement lines are used each statement must be separated by a colon (:) with the following exceptions, a colon is not required:

a) after an IF . . . . THEN statement, the THEN acts as an automatic separator.
b) after the final statement in a line, the carriage return, C/R, acts as the separator.

However, for consistency, a colon may be inserted after every statement, and will be ignored if it is not required.

# 4. CREATING A PROGRAM

The ORION data logger can only accept program input when there are no Tasks running. If a Task is running the ORION 'HALT' command, HA, must be sent before program lines will be accepted by the logger.

All program lines must begin with a line number, the first digit of which will identify the task with which the program will be associated, see section 4.1. It is this four digit line number that identifies to ORION that the input is to be interpreted as a program line, all lines not starting with a line number will be interpreted as an ORION command string.

## 4.1 ENTERING A PROGRAM

A program is entered by typing each statement line, starting with a line number and ending with a carriage return, C/R

Each ORION Task has a fixed allocation of line numbers for the program associated with that task:

ie T000 to T999

where T is the number of the task in the range 1 to 8.

**Note:** Within a program a branch can not be made to a line number outside those allocated to that task.

eg for a program associated with ORION Task 1 the statement

1050 GOTO 3105

would be illegal.

## 4.2 CORRECTING A LINE

A program line may be corrected by typing the replacement line in full, including the line number and carriage return, C/R.

## 4.3 DELETING A LINE

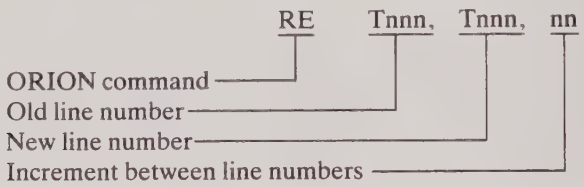
A line may be deleted from a program by typing the line number only, followed by carriage return, C/R.

## 4.4 INSERTING A LINE

A line may be inserted between two adjacent lines, line numbers permitting, by typing a suitable line number followed by the new program statement and carriage return, C/R.
If a suitable line number is not available the program may be renumbered.

## 4.5 RENUMBERING PROGRAM LINES

An entire program or part of a program may be renumbered by using the ORION command RE. The format of the REnumber command is:




Chapter 5, section 7 has further details on the ORION command REnumber.

## 4.6 LIST A PROGRAM

A processing program can be listed using the ORION LIst command. There are however certain limitations on the use of the command.

When using an ORION delta, 3530D, the list command may be used to list the program to the following interfaces:

1. The ORION delta printer PR
2. The ORION delta display DI
3. The ORION delta rear RS232 interface S0

For an ORION F,3530F,(without front panel keys) the list command may be used to list the program to the following interfaces:

1. The ORION F RS232 socket on the front panel DI
2. The ORION F RS232 interface socket on the rear S0

The LIST command is of the following form:

LI PR nnnn-mmmm TO DI

where nnnn-mmmm are the program line numbers to be listed.

**Note:** Use of interface options other than those given above will give unpredictable results.

Chapter 5, section 6, gives further details on the use of the ORION List command.

## 5. PROGRAM CONSTANTS

Table 6.1 shows the type of constants recognised by the ORION processing program.

| TYPE | EXAMPLE | DESCRIPTION |
|---|---|---|
| Numeric | 123.4567 | Numeric constants, up to 7 significant figures Decimal constants **must** have leading zeros included. |
| Exponential | -1.234E-15 | Exponential format constant, up to 4 significant figures in a 10 character field. This format is automatically selected for output by ORION delta if either:<br>a. 4 or less significant figures would be displayed, or<br>b. The value is greater than 1E8.<br>**Note:** except where the minus sign is used to indicate subtractions or a negative exponent, it must be preceded by a space. |
| Textural | "RESULT"<br><br>"PSI"<br>"OVERLOAD" | Text to accompany a processed result in Full format; can be up to 18 characters.<br>Units to accompany result; can be up to 3 characters.<br>A message, can be up to 38 characters in Full format and up to 18 characters in compact format. |

**Table 6.1 Constants**

## 6. PROGRAM VARIABLES

The ORION programming language allows the use of both simple and indexed variables.

The term variable is used to mean the name of a location or series of locations in ORION memory into which information may be written or from which it may be retrieved.

ORION variables may be either User defined or Predefined by ORION and are discussed in the sections that follow. Chapter 12 of the 3530 ORION Operating Manual contains a more detailed description of variables and their uses.

### 6.1 PREDEFINED VARIABLES

The predefined variables have had names permanently assigned to them as a design feature of ORION. Each of these variables has a specific function associated with ORION processing.

The ORION variables that deal with measured channel data are all **local** variables, ie the data is specific to the task with which the program is associated and may not be accessed directly from any other task.

All other variables are **global** and may be accessed from any task. They may be used to pass data or information between tasks.

Table 6.2 gives a list of the ORION predefined variables.

| Simple Variables | Usage | Global/Local |
|---|---|---|
| C1 thru C600 | Channel result (must be in scan set) | Local |
| H1 thru H600 | Channel high alarm limit | Global |
| L1 thru L600 | Channel low alarm limit | Global |
| COUNT1 thru COUNT8 | Scan counters for tasks 1 to 8 | Global |
| TIMER1 thru TIMER8 | Elapsed timers | Global |
| TIMER0 | Elapsed time from RUN, resetable | Global |
| **Indexed Variables** | | |
| C(I) | Channel I result | Local |
| H(I) | High limit of channel I | Global |
| L(I) | Low limit of channel I | Global |
| SC(I) | Ith channel result in scan set | Local |
| SH(I) | High limit of Ith channel in scan set | Global |
| SL(I) | Low limit of Ith channel in scan set | Global |
| SN(I) | Channel number of Ith channel scanned | Global |
| COUNT (I) | Indexed assess to counters 1 thru 8 | Global |
| TIMER (I) | Indexed assess to timers 1 thru 8 | Global |

**Table 6.2 ORION Predefined Variable**

A brief description of the ORION predefined variable follows, for more detailed information refer to the relevant section of the 3530 ORION Delta Operating Manual, Chapter 12.

**Channel Variables [C1 to C600, C(I)]**
Each ORION task has a set of channel variables associated exclusively with it in which the latest scan data for the task is held. These variables may be accessed using the simple variables C1 to C600 or by the indexed equivalent C(I).

Use of a channel variable that has not been defined as part of the Task channel set will result in an error message.

Processing may be performed on a channel variable and the result stored back in that variable.

eg 1011 C10=C10*5

However, any result, processed or not, will be overwritten by the new result on the next scan for that task.

**Channel alarm limit variable H1 to H600, H(I),L1 to L600, L(I)**
The alarm limit variables will initially contain the high and low alarm limit values set up during channel definition, see Chapter 3. Channels for which no alarm limits have been specified will contain a value of zero.

The alarm limit variables are Global, a change made to the high/low limit for a channel in one task program will affect any other task using that channel alarm limit.

**Channel Scan Set variable – SC(I), SH(I), SL(I)**
In addition to accessing the results in channel number order the channel results and limits may be accessed in the order in which they were scanned, ie; the order as specified in the task definition.

SC(I) – The channel scan set variable is a local variable, and the result is applicable only to the channel set associated with that Task.

SH(I), SL(I) – The scan set variables for the high and low limits are global variables. Whilst the use of SH(I) or SL(I) refers only to the (I)th channel in the scan set for that Task, changes made to the limits will globally affect all other Tasks using any or all of the channel limits affected.

**Channel number variable SN(I)**
The channel number variable, SN(I), allows the actual channel number of the Ith element in a scan to be determined. The variable will contain the channel numbers of the scan task in the order in which they were specified in the Task definition.

**Scan Counter Variables – COUNT1 to COUNT8, COUNT(I)**
There are eight scan counter variables, COUNT1 to COUNT8, that are permanently assigned to Tasks 1 to 8 respectively to keep a count of the number of times a task has been performed. Counters may neither be reset to zero or preset by the user. Each counter may be accessed using its indexed form COUNT(I).

Each scan counter will contain a value as follows:

| | |
|---|---|
| zero | If the task has not yet been triggered. |
| positive number | Representing the number of scans so far performed for the task, ie since run command was given. |
| negative number | Representing the number of scans completed, ie before the task was aborted, or when the requested number of scans has been reached (Note). |

**Note:** If a task is re-triggered the count will revert to positive and the count will continue from the previous total reached.

**Elapsed timer variable – TIMER0 to TIMER8, TIMER(I)**
There are nine elapsed timer variables, TIMER0 to TIMER8. The timer variables are not task dependent they are global variables and can be used in any task.

TIMER0 will contain the elapsed time from when the ORION 'RUN' command was given when it will automatically be reset to zero and started. TIMER0 can only be accessed by its simple variable name 'TIMER0' but can be accessed by any task and can be reset to zero using an assign command.

TIMER1 to TIMER8 are general timers which can be allocated for use by any task. They can be accessed using either the simple or the indexed form of the variables. Unlike TIMER0 they are not automatically started but require an assign command to start them:

```
eg      1010 TIMER3=0
```

However, the assign statement will not preset a start value, the timer will always start the count from zero.

```
eg      1011 TIMER4=20
```

will reset timer 4 to zero.

## 6.2 USER DEFINED VARIABLES

All user defined variables are global and can be used to pass scan task, or other information, that might be held in local variables between tasks.

The following rules apply when using user defined variables:

1. Variables must only be defined once, ie in the lowest numbered task program in which they are used.
2. The variable name must **always** start with an alpha character.
3. The name may contain up to a maximum of 8 alpha numeric characters.
4. The name may consist of alpha numeric characters only, ie no 'special' characters may be included.
5. Simple variables need not be declared before use.

   eg 1150 WATTS = C1 * C2

where WATTS is a simple variable into which the product of channels 1 and 2 will be stored.

6. User defined indexed variables **must** always be declared before use, at the start of a program, and are of the form: Line number, V iable; size.

   eg 1011 TOTAL; 5

where the indexed variable, TOTAL is declared as a five element array.

# 7 MATHEMATICAL OPERATORS AND FUNCTIONS

Table 6.3 lists the operators that are available for inclusion in an ORION processing program.

| Type | Symbol | Operation | Example |
|---|---|---|---|
| Mathematical | + | Addition | A + B |
| | − | Subtraction | A − B |
| | * | Multiplication | A * B |
| | / | Division | A / B |
| | ** | Integar Exponentiation | for $A^2$ type A**2 |
| Relational | < | Less than | A < B |
| | > | Greater than | A > B |
| | = | Equality | A = B |
| | <> | Non equality | A <> B |
| | < = | Less than or equal to | A < = B |
| | > = | Greater than or equal to | A > = B |
| Boolean | AND | Boolean and | A AND B = C |
| | OR | Boolean or | B = C OR A = C |

**Table 6.3 Program Operators**

The following standard mathematical function can be used when manipulating data:

| | |
|---|---|
| ABS(X) | Absolute value of X, eg +7.6 becomes 7.6<br>−7.6 becomes 7.6 |
| SQR(X) | Square root of X where X must be a value greater than 0 and a positive number. |
| SIN(X)<br>COS(X) | Trignometric functions, X must be in radians |
| ATAN(X) | Inverse trignometric function ie; arctan X where X is in radians |
| EXP(X) | Exponentiation ie; $e^x$ |
| LN(X) | Log of $e^x$ ie the natural log of x |
| INT(X) | Integar part of x eg +6.6. will become 6<br>−6.6 will become 7 |

The general format for inclusion of a function into a program statement is:

VARIABLE = FUNCTION (ARGUMENT)

for example the line:

3050 CHAN10 = SQR (C10)

would result in the square root of the result for channel 10 being stored in the variable CHAN10.

**Pending Operations, Parenthesis, etc.**
An expression can contain a maximum of four pending operations. The operations are performed according to standard mathematical rules. The expression are evaluated from left to right, ie as the processor finds a higher priority calculation the lower priority ones are left pending.

# 8. CONTROL STRUCTURES

## 8.1 DECISION MAKING – IF THEN ELSE ENDIF

To determine or alter the course of an ORION delta processing program the IF THEN statement may be used in conjunction with relational expressions. The simplest form of the statement would be:

| | | | | |
|---|---|---|---|---|
| **IF** | relational expression | **THEN** | statement | **ENDIF** |

Where the IF THEN statement causes the program to perform statement 1 only when the relational expression is true.

eg
```
1004 IF C1>MAX(1) THEN
1005 MAX(1)=C1
1006 ENDIF
```

where the expression C1>MAX(1) is evaluated and if true the statement MAX(1)=C1 is actioned, ie if the result from channel 1 is greater than the existing maximum value held for that Channel than the maximum, MAX(1) becomes equal to the result for channel, C1.

If however the expression is evaluated as false, C1<=MAX(1) then the statement in line 1005 is ignored.

However, by using the IF THEN ELSE form of the statement two alternatives may be set up for the same relational expression. The general form of the statements would be:

| | | | | | | |
|---|---|---|---|---|---|---|
| **IF** | relational expression | **THEN** | statement 1 | **ELSE** | statement 2 | **ENDIF** |

Where the IF THEN statement causes the program to perform statement 1 if the relational expression is true or statement 2 if the expression is false.

eg
```
1004 IF C1>MAX(1) THEN
1005 MAX(1)=C1
1006 ELSE
1007 PRINT "CHANNEL 1 STABLE"
1008 ENDIF
```

Where if the expression C1>MAX(1) is evaluated as true line 1005 is actioned and lines 1006 and 1007 are ignored. If however, the expression is evaluated as false line 1007 is actioned.

**Nested IF's**
The IF THEN structure is nestable up to four levels, however the user should note that there must be one ENDIF statement for every IF statement used.

## 8.2 PROGRAM LOOPS – FOR . . . NEXT

A program loop is a procedure in which one or more program operation is repeatedly executed. The ORION delta processing instructions that define the loop are the FOR . . . NEXT statements. The user is able to specify how many times the loop is to be repeated by using the TO and STEP commands.

The simplest form of the loop statement is:

FOR I = x TO y : statement : NEXT

where: I is the loop variable
x is the initial value assigned to I
y is the final value assigned to I

The statement FOR identifies the start of the loop, the statement NEXT causes I to be set to x+1 and the instructions within the loop repeated.

eg To print the results for a channel set of 50 channels

```
3005 FOR I=1 TO 50
3006 PRINT SC(I)
3007 NEXT
```

The loop will be repeated until I has been set to 50 when instruction 3006 will be carried out for the last time.

**Loop Variables** The loop variable, I, used in the example is a global variable and therefore any value assigned to it in one task will apply if it is used in another task.

Although I is the "textbook" name generally given to a loop variable it is a simple variable for which any valid simple variable name may be used, eg Y, I1, LOOP1, etc.

**Note:** Predefined variable names must **not** be used, eg C, H, L etc.

**N.B.** Once a simple variable has been declared as a "loop variable" it cannot have a fixed value assigned to it in its own or any other task.

**STEP** The STEP function adds flexibility to a FOR . . . NEXT loop and can be included as follows:

FOR I = x TO y STEP z: statement: NEXT

where: z is a positive or negative integer, and is the increment by which the initial value will be increased or decreased.

If STEP is not included the incremental value is assumed to be 1, as in the previous example. However the example could be modified to print every second channel result by modifying line 3005.

eg
```
3005 FOR I = 1 TO 50 STEP 2
3006 PRINT SC(I)
3007 NEXT
```

**Nested Loops**
FOR . . . NEXT loops are nestable up to four levels. It should be remembered however that the FOR . . . NEXT statement must always be paired, so each FOR statement must always be terminated by its own NEXT statement.

## 8.3 UNCONDITIONAL BRANCH – GOTO

Unconditional branches are written using the GOTO statement in the form:

GOTO Tnnn

where: T is the task number of the task which contains the program.
nnn is a line number within the task program.

The GOTO statement will cause the program to branch to the line number indicated in the statement and continue program execution from that line number.

Branches may be in a forward or backward direction through a program, but must be the last statement on a multi-statement line.

# 9. THE OUTPUT STATEMENTS

Results or processed data, other than that held in the channel variables and governed by the Task definition, will not be output to any interface or device unless the processing program contains an output statement.

The two main output statements are the PRINT and WRITE statements, and can be used to send data or information to an output device.

1. PRINT statement, will add its information to the Task data output queue, ie the information will always be output within the confines of the Start and Done Task markers. It is generally used to output result or processed data.

2. WRITE statement, will cause the information to be output to the selected device immediately it is encountered in the program. It is generally used for sending messages.

The two other forms of the output statements are KILL and SET . . . TO.

3. KILL, is used to delete redundant Channel results from the output data.

4. SET . . . TO, is used to control the loggers analogue and digital outputs.

A brief description of each type of statement is given in this section. More detailed information will be found in Chapter 12 of the ORION delta Operating manual.

## 9.1 THE PRINT STATEMENT

The action of the PRINT statement is to add a processed result to the current scanned Channel results. The processed result will appear within the confines of the Start and Done Task markers.

The general form of the complete PRINT statement with extensions is:

PRINT [modifier] [device] item;[extensions], [item;[extensions]] . . . etc.

where the square brackets [ ] indicate an optional parameter.

**Item**
The simplest form of the PRINT statement is one without extension or other optional parameters ie:

PRINT item

and will usually take the form of a), b) or c) below.

a) PRINT expression. This form of PRINT statement causes a processed result, in non-alarm status, to be appended to the current scan **after** the Channel results but **before** the Done Task marker. The result will have no text or units, and will be known as processed result number 1.

eg 1005 PRINT MEAN

b) PRINT indexed variable. This statement causes a result to be appended as above, using a previously declare indexed variable.

eg 2010 PRINT CHAN(10)

c) PRINT "string". This statement causes a "text only" result to be inserted **before** any existing results in the scan, but **after** any other "text only" results. Up to 18 characters may be used for compact format and up to 38 for full format output.

eg 5011 PRINT "CHANNELS 1 TO 15"

**[modifier]**
The modifier is used to force the output into alarm format when a processed result is outside the permitted limits. The results will usually form part of a conditional structure, eg an IF . . . . THEN statement, in order to determine the Alarm status. The modifiers are:

< which forces low Alarm status
> which forces high Alarm status

eg 5015 PRINT > CHAN(10)

In the above example, assuming the Task definition had specified Full format output, the result for CHAN(10) would be output in the form:

! P 001 15.125 >.

Alarm text could be included in the output either using the Print extensions, as described in section 9.2, or the indexed variable CHAN(10) could have been defined using the Output Format declaration, as described in section 9.3, in which case the alarm text would be added automatically.

**[device code]**
The device code consists of a number of codes, which direct the processed results to a particular output device. The code need only be included in the statement if the output device required for the specific result is different to that specified for the associated Task.

| **Action** | **Device** |
|---|---|
| Print to Display | \ DI |
| Print to RS232 | \ S0 |
| Print to Printer | \ PR |
| Print to Cartridge | \ CA |
| Print to GPIB | \ GP |
| Print to RS422/a | \ S1 |
| Print to RS422/b | \ S2 |

By adding the device code to the previous example the alarm output could be directed to another device.

eg 5015 PRINT > \GP CHAN(10)

would direct the output to the GPIB interface.

## 9.2 PRINT STATEMENT EXTENSIONS

The previous section described the basic form of the PRINT statement, ie one without the use of print extensions.

PRINT [modifier] [device] item

However, with the exception of PRINT "string" the PRINT statement may be extended to make selected items of output information more meaningful, and to enable alarm text to accompany processed results.

PRINT [modifier] [device] item [;extension] [;extension] [;extension]

**Note 1:** Each extension must be separated by a semicolon (;).

The PRINT statement may be extended to contain the following items of information in the combinations given:

a) Units
b) Processed result number
c) Processed result number and units
d) Text and processed result number
e) Text, processed result number and units

**Note 2:** When using a PRINT extensions the output will remain in the standard ORION output format, as described in Chapter 7.

PRINT extensions are of particular use with simple variables, ie to add explanitory or alarm text to output. However, caution must be used when using PRINT extensions with indexed variables which have been defined using the Output Format Declaration. When a PRINT extension is used the Output Format Declaration will be completely overridden, and if not replaced by an extension item the field will remain blank.

### 9.2.1 Change Processed Result Number

All processed results will be numbered P001 unless otherwise defined. The print extension can be used to change the processed result number to one more appropriate. The semicolon must be used followed by the required result number.

eg 1011 PRINT CHAN(10);10

will produce output in the form:

: P 010 5.435

assuming output has been requested in Full format.

### 9.2.2 Adding Units

Units of up to three characters can be added to the processed results. The semicolon separator must be used and the units must be enclosed in inverted commas.

eg 1011 PRINT CHAN(10);10;"VDC"

will produce output in the form:

: P 010 5.435 VDC .

### 9.2.3 Adding Descriptive text

Text of up to 18 characters may be added to the output in Full format. Text must be enclosed in inverted commas and preceeded by the semicolon separator.

eg 1011 PRINT CHAN(10);"NORMAL";10;"VDC"

will produce output in the form:

:NORMAL P 010 5.435 VDC .

By including the modifier in the print statement the text may be used as alarm text.

eg
```
1010 IF CHAN(10) < = 5, THEN
1011 PRINT CHAN(10);"NORMAL";10;"VDC"
1012 ELSE
1013 PRINT > CHAN(10);"OVERLOAD";10;"VDC"
1014 ENDIF
```

in which case if CHAN(10) is not less than or equal to 5 the alarm text will be printed.

!OVERLOAD P 010 6:015 VDC > .

## 9.3 OUTPUT FORMAT DECLARATION

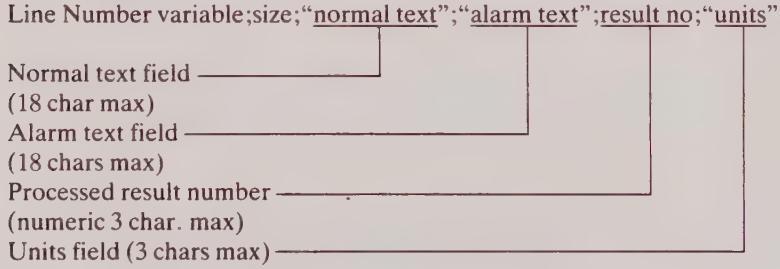
As stated in section 6.2, User Defined variables, all indexed variables should be defined at the start of the program, and take the general form:

Line number variable; size

where: variable is the variable name
size is the number of elements in the array
; the semicolon separator is mandatory

However, the declaration can be extended to contain all or part of the additional information that could normally be found in a Full format ORION result output. This extended variable definition is referred to as an Output Format Declaration and takes the following general form.

Line Number variable;size;"normal text";"alarm text";result no;"units"




**Note 1:** a) Semicolon separators must be included between fields.
b) All text for printing must be enclosed in inverted commas.

The complete declaration can be reduced to the following combination of items:

a) Processed result number
b) Text
c) Processed result number and units
d) Text and processed result number
e) Text, processed result number and units

**Note 2:** If "normal text" is to be omitted but "alarm text" is to be entered the inverted commas must still be included as shown below.

Tnnn variable;size;"";"alarm text"

**Text**
The text included in the declaration will apply to all elements of the array. However, by using a PRINT statement with extensions the Output Format Declaration for all the elements in the array can be overridden.

```
eg      1000 POWER;10;"";"OVERLOAD";1;"WAT"
        .
        .
        .
        1015 PRINT POWER(5);"NORMAL";
```

will produce the output:

```
: NORMAL              P 001  30.67
```

**Result Number**
The result numbers can be in the range of 1 to 999. The number defined in the declaration will be the number given to the first element in the array, all other elements will be allocated numbers following on in sequence.

## 9.4 THE WRITE STATEMENT

The WRITE statement enables **text messages** to be sent to the output device specified in the Task Definition. A device code may be included to send the message to a different output device.

The main differences between the Print and Write Statements are that the latter:

a) Are treated as text only, although this text may include a calculation, and are not in ORION result format.
b) Do **not** appear within the relevant Start and Done Task markers. They are output as soon as possible after the processor has encountered them in the program. Their precise position in the sequence of output data is difficult to predict, owing to the multi-tasking nature of the system software, but it is likely that the message will appear **before** the standard scan output.

The general form of the write statement is:

WRITE [device code]  item,item,.....item.

**[device code]**
The device code defines the output device to which the message is to be sent and need only be included in the statement if it is different from that specified in the associated task definition. The device codes are as follows:

| Action | Device Code |
|---|---|
| Write to Display | \DI |
| Write to Printer | \PR |
| Write to Cartridge | \CA |
| Write to GPIB | \GP |
| Write to RS232 | \S0 |
| Write to RS422/a | \S1 |
| Write to RS422/b | \S2 |

Only one device code may be included in any one WRITE statement, however the message may be sent to several output devices by duplicating the WRITE statement but changing the device code in each.

**Items**
The items may be one of two types:

a) An expression, ie any variable or mathematical or Boolean expression that evaluates to a numeric result, eg C1*10.
b) A "string" which may be any string of characters enclosed in quotation marks, eg "RESULT".

A write statement may include several items as shown earlier, each item must be separated from the next by a comma. This comma causes each item to be added to the output message at specific tab positions in the standard 80 character output format, as shown below.

```
00....11....22....33....44....55....66....77..
tab   tab   tab   tab   tab   tab   tab   tab
```

If the item is an expression, ie numeric value, it is limited to the numeric field size of ten numeric characters and will be followed by a space character. Each numeric item will begin on a tab position.

If the item is a "string" it is not limited to a specific number of characters, other than by the input constraints laid out in section 3.2. It will occupy the exact number of characters specified. However, the next item in the WRITE statement will begin at the next available tab position.

```
eg      2010    WRITE  \DI "VOLTAGE OVERLOAD AT", TIME

        VOLTAGE OVERLOAD AT     10.30
        0                11     22
        tab              tab    tab
```

In addition to sending alarm and warning messages to an output device whilst a Task is running the WRITE statement can be effectively used to output results in the form of a table, as shown in the following example, by making full use of the tab positions.

```
eg      1003 WRITE  \S0 "WATER BATH STATUS (DEG C)"
        1004 WRITE  \S0 "TOP", "BOTTOM", "VESSEL", "MEAN"
        1005 WRITE  \S0 C3, c1, C2 MEAN
```

would produce output in the form:

```
WATER BATH STATUS (DEG C)
TOP          BOTTOM       VESSEL       MEAN
20           12           13           15
```

## 9.5 THE KILL STATEMENT

The KILL statement is used to delete Channel results from the output data stream sent to the output device.

The general form of the statement is:

KILL p,q,r, ..... etc.

where p,q,r, .... are channel numbers, each number must be separated by a comma.

The KILL statement is applicable only to the pre-defined variable C1 to C600, once a channel result has been killed it cannot be accessed later in the program. The killed channel result storage space is made available for reuse immediately the statement has been executed.

In a KILL statement only the element number must be used, eg to remove the results for channel 1,2 and 3 write:

KILL 1,2,3

**not** KILL C1,C2,C3

To kill consecutive channel numbers a FOR...NEXT loop may be used and the indexed form of the element number:

1010 KILL(I)

To KILL a non-consecutive sequence of channel numbers in a channel set the Scanned Channel number form:

2015 KILL SN(I)

may be used.

## 9.6 SET...TO STATEMENTS

The SET...TO statement can be used to influence both the analogue and digital outputs of the logger.

**SET analogue channel TO value.** This instruction sets the output of the specified Analogue Channel to the value defined, where the value defined represents a percentage of full scale.

eg 1020 SET 81 to 50

will set analogue channel 81 to 50% full scale.

The SET...TO instruction can either set the output to a fixed value, or cause it to follow the values of one of the program variables.

**SET digital channel TO value.** Digital outputs can only be in one of two possible states, viz: "zero" and "non-zero". These states are usually identified by "0" and "1" respectively. The actual value of the 1 output can be selected by the user. Any value specified other than zero will set the output to 1.

eg 1021 SET 42 TO 0

**SET digital channel TO "string"** Several consecutive Digital Channels can have their values included in a "string" instruction.

eg 1030 SET 45 TO "1010"

will set channels 45 to 48 to 1010 respectively.

**Notes:** 1. Only the first Channel in the sequence must be specified in the instruction.
2. The inverted commas around the strings are essential.
3. Only constants may be included in the string.

## 10. DIGITAL INPUTS

Processing can be performed on digital signals, either as single bit inputs or as groups.

**Single Bit Inputs**
Each singel bit input is regarded as having the value of either 1 or 0. The two Boolean operators AND and OR can be used in the IF.....THEN statements.

eg 2005 IF C10=1 AND C12=1 THEN

where the program will take different paths depending on whether the expression is true or false.

Or alternatively using an equation.

eg 2005 RESULTS=C10=C12=1

where the equation C10=C12=1 can be either true or false giving RESULT the value of 1 if true.
0 if false

**Grouped Inputs**
Digital inputs can be grouped together in the Status/Event Channel Definition, see Chapter 3. The grouping process causes these inputs to be regarded as a decimal number.

eg A group of four channels C10 to C13 requiring an input of

| C13 | C12 | C11 | C10 |
|---|---|---|---|
| 1 | 0 | 1 | 1 |

can be regarded as the decimal number 11 (8+2+1) the program line could read:

3009 IF C10=11 THEN.....

## 11. CONTROL OF OTHER TASKS

It is possible to control the activity of other Tasks by means of flexible cross-triggering and cross-aborting statements, which may be included within IF . . . . THEN control structures.

### 11.1 TRIGGERING OTHER TASKS

TRIGGER x, y, z. This statement will cause Tasks x, y and z to be triggered. The trigger will simple be ignored if the Task is already active. Up to all 8 Tasks may be specified in the Task list, although self-triggering is of little practical use.

### 11.2 ABORTING OTHER TASKS

ABORT p,q,r. This statement will cause Tasks p, q, and r to be aborted, ie stopped. Up to all 8 Tasks may be included in the Task list, and self-aborting is acceptable.

**Note:** The list of tasks to be triggered or aborted may be entered in any numerical order, however, ORION sorts the task numbers into ascending numerical order and if the program line is subsequently listed or recalled on the front panel display that is how the tasks will appear.

## 12. COUNTERS AND TIMERS

The ORION counters and timers have already been discussed in section 6.1 under Program variables – Pre-defined. This section however contains further information on their uses and limitation.

### 12.1 COUNTERS COUNT1 TO COUNT8

As previously stated each Task has its own dedicated counter which can be interrogated but not modified by the processing program.

Each counter is automatically set to zero each time the RUN command is given and will keep a count of the number of scans that its associated task has performed since it was first triggered.

The counter is updated at the beginning of a scan and therefore, if a counter is to be used by a processing program, it should be noted that calculation errors could occur if the interval between scans is small. This is because it is possible that a second scan may have started, and the counter updated, before the first had been fully processed.

### 12.2 TIMERS TIMER0 TO TIMER9

As previously stated TIMER0 will be automatically set to zero and started when the RUN command is given. However, TIMERs1 to 8 may be assigned by the user to any task, and must be started by assigning a value to them.

The timers count in seconds with a resolution of $1/10^{TH}$ of a second.

If the user wishes to use a timer in calculations within a processing program it is advisable to "freeze" the time before use, by writing for example:

```
TIME=TIMER4
```

and using the value in TIME for the calculation.

## 13. RANDOM CHANNEL ACCESS

Where the channels scanned in a Task are always the same and/or scanned consecutively then use of the channel variables of form C1 or C(I) can be efficient and satisfactory.

However, in some Tasks the scan set may have been defined in a non-consecutive order or, although the number of channels to be scanned and the processing performed on them may be constant, the actual channels in the scan set may vary. In these circumstances using the Scanned Channel variable as described in section 6.1 will greatly simplify the processing program, ie by using the Scanned Channel variables the user will always have access to the Ith channel of scanned set.

| | |
|---|---|
| SC(I) | Ith Channel result in scan set |
| SH(I) | High limit for Ith channel in scan set |
| SL(I) | Low Limit for Ith channel in scan set |
| SN(I) | Gives Channel number of Ith channel scanned |

The following is an example of how the channel set variable and the scanned channel number could be used when the number of channels in a Task channel set remains constant but the actual channel numbers assigned to that set vary.

If Task 2 is defined with the channel set 1, 3, 6, 7, 10 and the result for each channel is to be printed together with its channel number. The lines:

```
2005 FOR I = 1 TO 5
2006 PRINT SC(I);SN(I);"PSI"
2007 NEXT
```

would produce an output, assuming compact format has been selected, of:

```
S T 2  05:10:10.7
P 001  1.991     PSI .
P 003  2.000     PSI .
P 006  1.999     PSI .
P 007  2.000     PSI .
P 010  2.000     PSI .
D T 2
```

However changing the channels in the channel set to 3, 6, 10, 11, 12 would not affect the running of the program but would produce a print out of:

```
S T 2  07:10:13.5
P 003  2.000     PSI .
P 006  1.999     PSI .
P 010  2.000     PSI .
P 011  1.998     PSI .
P 012  1.999     PSI .
D T 2
```

High and low limit checks can easily be included into such a program using the SH(I) and SL(I) variables.

eg
```
2005 FOR I=1 TO 5
2006 IF SC(I) >SH(I) THEN
2007 PRINT > \PR SC(I);SN(I);"PSI"
2008 ENDIF
2009 PRINT SC(I);SN(I);"PSI"
2010 NEXT
```

where by the inclusion of lines 2006 to 2008 the result in alarm format will additionally be displayed on the ORION printer.

In general the Scanned Channel variables can normally be used in place of the channel equivalent without affecting the program operation.

## 14. MEMORY CAPACITY

The amount or ORION free memory available for the user processing program is dependent on the number of input/output modules connected to ORION. With the maximum of 600 channels fitted, ie 200 channel ORION with two 200 channel slave units there are 8,976 bytes of free memory available for user programs. This is equivalent to approximately 100 lines of user program.

Each occupied slot of the ORION or slave unit, ie slots 4 to 13, occupy 520 bytes of memory. Memory can be freed for program use by removing cards from the slots. However, slots **must** be depopulated from the top, ie the highest channel numbers, in order to release memory. Removing an intermediate channel card will **not** release memory.

Table 6.4 gives examples of the free memory available for various configuration of ORION.

| ORION | CHANNEL No.'s | Free Memory | Program Lines |
|---|---|---|---|
| ORION with 2 slaves | 1-600 | 8,976 bytes | ≃ 100 |
| ORION with 2 slaves | 1-500 | 11,576 bytes | ≃ 130 |
| ORION with 1 slave | 1-400 | 14,176 bytes | ≃ 160 |
| ORION only | 1-200 | 19,376 bytes | ≃ 220 |
| ORION only | 1-120 | 21,456 bytes | ≃ 240 |

**Table 6.4 Memory capacity.**

### 14.1 PROCESSING PROGRAM MEMORY REQUIREMENTS

The amount of memory that a processing program will occupy will depend on two factors:

1. The number of programmer defined simple and indexed variables and the number of elements in each array.

2. The number of program statement lines.

When writing large programs it may be necessary to calculate the amount of memory that will be required by each program in order for the programs to run.

1. Each program line will always occupy 88 bytes of memory.

2. A SIMPLE variable will occupy 6 bytes + length of variable name

   eg TOTAL will occupy 6+5 = 11 bytes.

3. An INDEXED variable will occupy 12 bytes + 4xn bytes + length of variable name, where n is the number of elements in the array.

   eg AVERAGE (100) will occupy
   12+(4x100)+7 = 419 bytes

If program memory should become critical it may be necessary to use multi-statement lines in the program as each program line will always occupy 88 bytes of memory, regardless of the number of statements on that line.

## 14.2 STATEMENT MEMORY ALLOCATION

Each statement of an ORION program, when loaded into ORION memory, will occupy a varying number of memory locations depending on its component parts. Table 6.5 gives details of the byte allocation for each of these components.

Using the table the user can, when program memory is very short, calculate how many statements can be included in each line of the program. However the user must remember that each line will always occupy 88 bytes of memory.

| TYPE | Number of Bytes | Example | |
|---|---|---|---|
| Line Number | 4 | 1001 | (4 bytes) |
| Variable | 4 + length of name | AVE | (4+3=7 bytes) |
| Constant | 5 | 4.5 | (5 bytes) |
| Operator | 3 | + | (3 bytes) |
| Function | 3 | ATAN | (3 bytes) |
| Statement | 3 | PRINT | (3 bytes) |
| Text | 1 byte per character | TEXT | (4 bytes) |
| Separators/delimiters " ; : , | 1 byte each | "TEXT"; | (7 bytes) |
| Brackets ( ) | 1 byte each | (5) | (3 bytes) |

**Table 6.5 Statement memory allocation.**

The example that follows shows how much space may be saved by using multi-statement lines.

| eg | | Line No. | | Code | | Total |
|---|---|---|---|---|---|---|
| | 1020 IF C1>THEN | 4 | + | 15 | = | 19 |
| | 1021 SET 41 TO 1 | 4 | + | 16 | = | 20 |
| | 1022 ELSE | 4 | + | 3 | = | 7 |
| | 1023 SET 41 TO 0 | 4 | + | 16 | = | 20 |
| | 1024 ENDIF | 4 | + | 3 | = | 7 |

The above five lines of code will occupy 5 × 88 bytes of ORION memory ie 440 bytes.

However if all five statements are entered on one line:

```
1020 IF CI>5 THEN SET 41 TO1:ELSE SET 41 TO 0:ENDIF
```

This will only occupy 88 bytes of memory of saving of 352 bytes. A considerable saving if memory space is at a premium!

Program lines containing large calculations may have to be split onto more than one line however.

```
eg      8014 A=K1(I)*EN(I)+K2(I)*E1(I)    = 73 bytes

        8015 A=A+H3(I)*E2(I)              = 52 bytes
```

## 15. SPEED OF EXECUTION

If speed of processing is important then, when writing a program, the following information can be used to estimate the processing time of the program.

Allow:

**1ms** for each program line

In addition allow:

**1ms** for each arithmetic operation, indirect access function, assignment etc.
**5ms** for each transcendental function ie ABS, COS etc.

**NB:** When estimating processing time do not forget to evaluate FOR . . . NEXT loops appropriately.

## 16. PROGRAMMING TECHNIQUES

The information included in this section is designed to help the user when writing processing programs for his ORION. Important points that have already been mentioned in the chapter are re-emphasised in this section.

1. Always plan your processing program away from your machine, whether it is to be entered from the ORION front panel or via a remote interface.

2. Start line numbering from line T000, where T is the task number. Use consecutive line numbers, avoid the use of line numbers not followed by a declaration or statement.

3. Declare indexed variable at the start of the program, use descriptive names for each variable.
   **Note:** Variables are always initialised, cleared to zero, when 'RUN' is given. However, if a program is aborted and then retriggered the variables will **not** be re-initialised.

4. Declare indexed variables that are to be accessed by more than one Task program in the program for the lowest numbered Task in which they are used.
   **NB:** Indexed variables must only be defined once.

5. Suffix commonly used variables, eg I, J etc., with Task numbers, eg I1, J2. This will help avoid accidental use of the same variable name in more than one task.

6. Use single statement lines, when program space allows, to make the program easy to understand or change. However if program space is critical always use multi-statement lines.

7. Use 'first scan processing', by devoting a small section at the start of the program to actions that are only to be performed the first time the program is actioned, processing time can be reduced. Use first scan processing to:

   a) Assign initial values to simple variables and elements of indexed variables.
   b) Perform any once only calculations.

8. Use the ELSE form of the IF . . . THEN statement whenever meaningful.
   To increase effectiveness of IF THEN statements use with BOOLEAN expressions.

   eg IF C1>C2 AND C3<C5 THEN
   .....
   ELSE

9. Manipulate data using the channel variables C(I) and SC(I) rather than introducing temporary variables. This will save on program memory.

10. Use the scan channel variables SC, SH, SL and SN to access channel or alarm data, in preference to the channel variable, C, and alarm variables H and L. This simplifies external adjustment of channel sets, ie the actual channel, numbers are irrelevant, providing the number of channels in the channel set does not change.

11. Use the KILL statement, when channel data is no longer required for processing or output, to free buffer space. The buffer space will be freed immediately the KILL command is actioned. In addition to freeing buffer space post processing will be speeded up, as alarm and checking and post processing is not carried out on channels that have been killed.

12. Save memory by overwriting existing variables instead of creating new ones.

13. Economise on statements by including expressions in PRINT and SET statements.

14. Use the channel set alarm limits to save assignments and variable space. Where possible use the ORION alarm checking and conversion facility as this is faster than using the processing program.

15. Avoid programs that generate a large number of processed output, ie using the PRINT statement, as this could cause buffer exhaustion.

16. Avoid use of WRITE statement for standard Task output, use WRITE for:

    a) Sending messages to other devices
    b) Alarm messages
    c) Producing information in tabular form

17. Avoid GOTO's.

18. Do not attempt accurate use of the timers. If a timer is to be used store the time to 'freeze' it before it is used in calculations etc.

19. Do not use the task counters for calculation, keep a count within the processing program it is more likely to be accurate, as the processing time can overlap into the next scan.

## 17. VOCABULARY OF FUNCTIONS

ORION Delta can recognise the words and symbols shown in the following alphabetical list. Each term is explained in this chapter in the section or sections indicated. Further information will be found in Chapter 12 of the 3530 ORION delta operating manual.


| TERM | SECTION |
|---|---|
| ABORT | 11.2 |
| ABS(x) | 7 |
| ATAN(x) | 7 |
| C1 to C600 | 6.1 |
| C(I) | 6.1 |
| Constants of the following types: | |
| Exponential | 5 |
| Messages | 9.4 |
| Numeric | 5 |
| Text | 5 |
| Units | 9.2, 9.3 |
| COS(x) | 7 |
| COUNT1 to COUNT8 | 6.1, 12.1 |
| COUNT(I) | 6.1 |
| ELSE | 8 |
| ENDIF | 8.1 |
| EXP(x) | 7 |
| FOR | 8.2 |
| GOTO | 8.3 |
| H1 to H600 | 6.1 |
| H(I) | 6.1 |
| IF | 8.1 |
| Indexed Variable | 6.1, 6.2, 9.3 |
| INT(x) | 7 |
| KILL | 9.5 |
| L1 to L600 | 6.1 |
| L(I) | 6.1 |
| LN(x) | 7 |
| NEXT | 8.2 |
| Operators of the following types: | |
| Boolean | 7 |
| Integer Exponentiation | 7 |
| Mathematical | 7 |
| Relational | 7 |
| Parentheses ( ) | 7 |
| PRINT | 9.1, 9.2 |
| SC(I) | 6.1, 13 |
| SET | 9.6 |
| SH(I) | 6.1, 13 |
| SIN(x) | 7 |
| Simple Variable | 6.1, 6.2 |
| SL(I) | 6.1, 13 |
| SN(I) | 6.1, 13 |
| SQR(x) | 7 |
| STEP | 8.2 |
| THEN | 8.1 |
| TIMER0 | 6.1, 12.2 |
| TIMER1 to TIMER8 | 6.1, 12.2 |
| TIMER(I) | 6.1 |
| TO | 8.2, 9.6 |
| TRIGGER | 11.1 |
| WRITE | 9, 9.4 |

# Chapter 7
# Output and Recording Formats


| *Para. No.* | *Title* | *Page No.* |
|---|---|---|
| 1 | General | 7.2 |
| 2 | Output Data Formats | 7.2 |
| 2.1 | Numerical Result | 7.3 |
| 2.2 | Start and End of Scan Markers | 7.3 |
| 2.3 | Event Result | 7.4 |
| 2.4 | Run and Halt Markers | 7.4 |
| 2.5 | Power Fail and Power Back Messages | 7.5 |
| 2.6 | Date | 7.5 |
| 2.7 | Specific Formatting | 7.6 |
| 3 | Coded Binary Format | 7.6 |
| 3.1 | Numerical Result | 7.8 |
| 3.2 | Start and End of Scan Markers | 7.11 |
| 3.3 | Event Result | 7.11 |
| 3.4 | Run and Halt Markers | 7.12 |
| 3.5 | Power Fail and Power Back | 7.12 |
| 3.6 | Date | 7.13 |
| 4 | Recording Data on Cartridge (3530D only) | 7.13 |
| 4.1 | Recording Modes | 7.13 |
| 4.2 | Recording Formats | 7.13 |
| 4.3 | Data Output Format | 7.14 |
| 4.4 | ORION Tape Program | 7.14 |
| 4.5 | The Cartridge Tape Format | 7.15 |

## 1. GENERAL

The user may select the format in which data, ie messages, measured results, markers etc., will be output by ORION. Three output formats are available: Full, Compact and Coded Binary format.

**Full:** 40 character ASCII string, contains full information including text messages defined by the text command. Primarily designed for display at a VDU or printer.

**Compact:** 20 characters ASCII string, contains full numeric data (results, channel nos. etc.) and ORION generated messages, but does not include text defined by the text command.

**Coded Binary:** 10 character ASCII string contains numeric data in binary form, encoded as printable ASCII characters. No text of any type is included. This format is intended primarily for recording data on a 3530D cartridge or other data collection device connected to one of the interfaces.

**Note:** When processed data is recorded onto a 3530D cartridge each processed data record may be followed by a further 10 character record containing information which will allow the ORION to reconstitute the processed data if replayed from cartridge. Section 4.2 of this chapter gives further details of binary coded processed data.

When the output format is not specified the ORION will default to output in Full format.

## 2. OUTPUT DATA FORMATS

The data output by ORION is dependent on logger action and can be grouped into six types as follows:

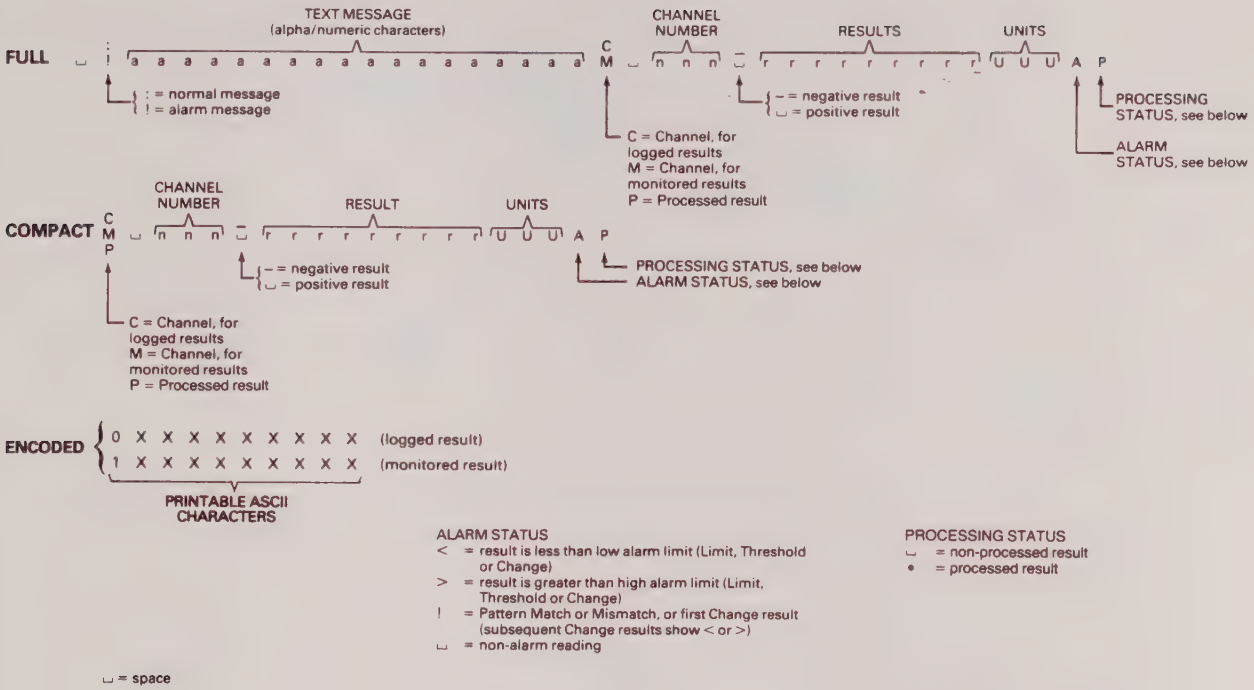
1. Numerical results.
2. Start and end scan markers.
3. Event results.
4. Run and Halt markers.
5. Power Fail and Power back messages.
6. Date.

The first character of each data string identifies the output format of that string, ie:

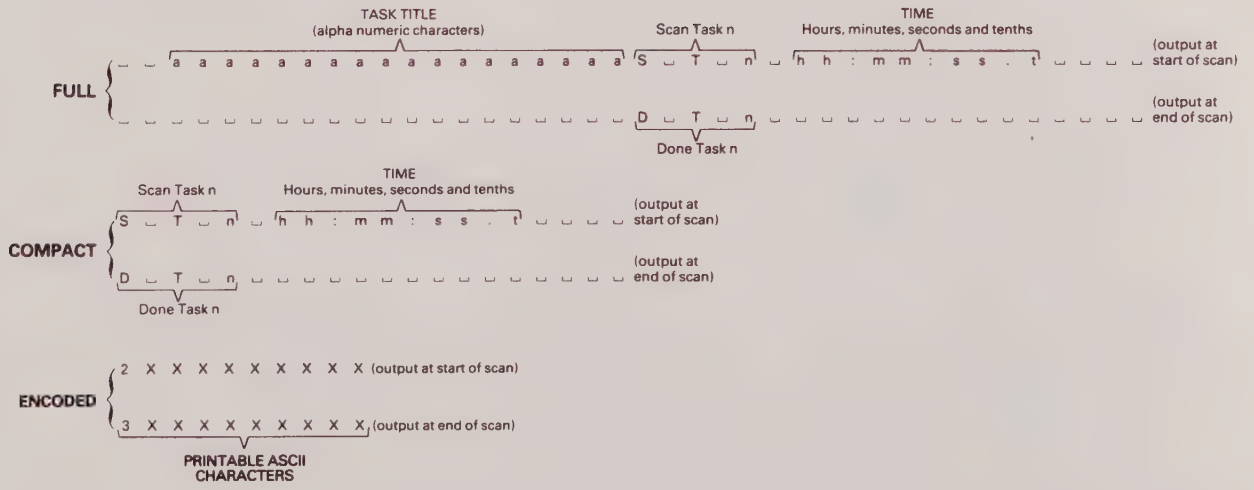
Space character : full format.
Alpha character : compact format.
Numeric character : encoded format.

This section gives details of the standard layout of each of the data types. Section 3 describes how the binary data is encoded to produce a Coded Binary format record.

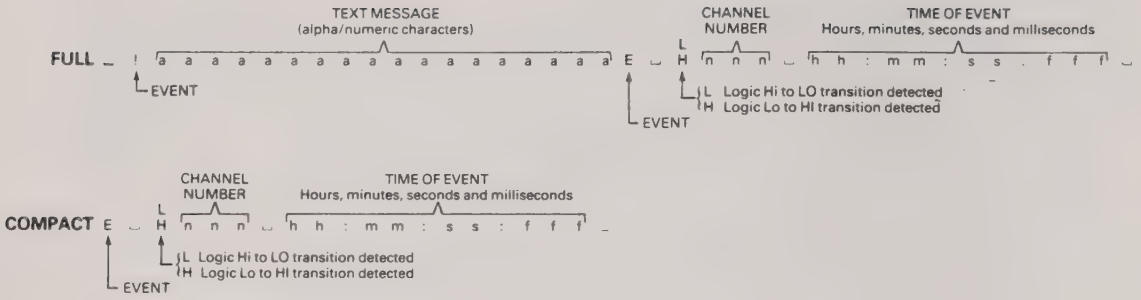
## 2.1 NUMERICAL RESULT


FULL
TEXT MESSAGE
(alpha/numeric characters)
CHANNEL NUMBER
RESULTS
UNITS
: = normal message
! = alarm message
C = Channel, for logged results
M = Channel, for monitored results
P = Processed result
– = negative result
␣ = positive result
PROCESSING STATUS, see below
ALARM STATUS, see below
COMPACT
CHANNEL NUMBER
RESULT
UNITS
– = negative result
␣ = positive result
PROCESSING STATUS, see below
ALARM STATUS, see below
C = Channel, for logged results
M = Channel, for monitored results
P = Processed result
ENCODED
0 X X X X X X X X X (logged result)
1 X X X X X X X X X (monitored result)
PRINTABLE ASCII CHARACTERS
ALARM STATUS
< = result is less than low alarm limit (Limit, Threshold or Change)
> = result is greater than high alarm limit (Limit, Threshold or Change)
! = Pattern Match or Mismatch, or first Change result (subsequent Change results show < or >)
␣ = non-alarm reading
PROCESSING STATUS
␣ = non-processed result
• = processed result
␣ = space

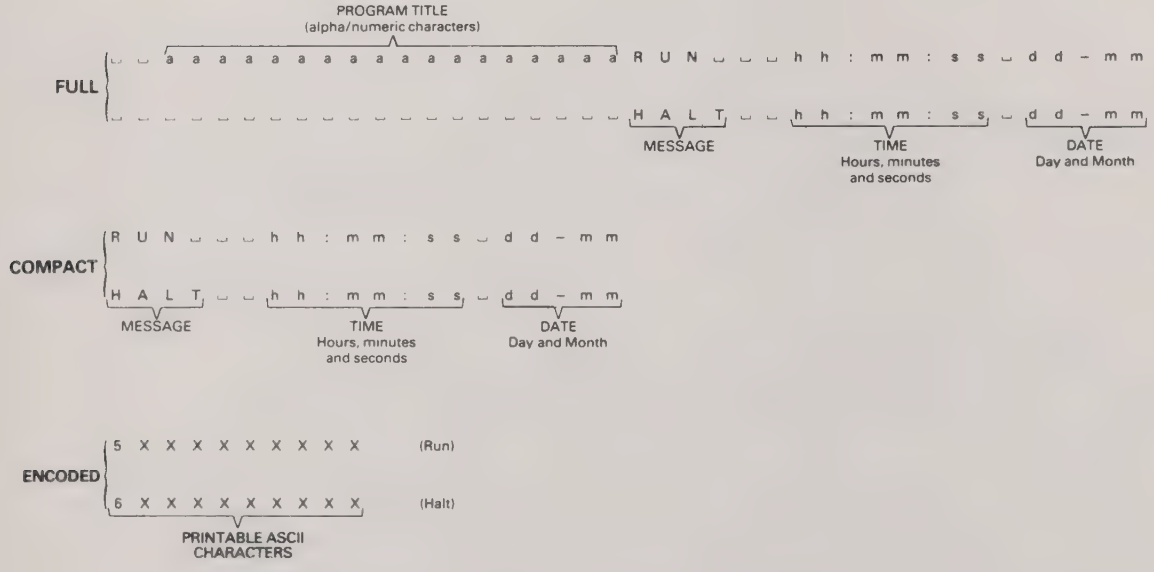
## 2.2 START AND END OF SCAN MARKERS


FULL
TASK TITLE
(alpha numeric characters)
Scan Task n
TIME
Hours, minutes, seconds and tenths
(output at start of scan)
(output at end of scan)
Done Task n
COMPACT
Scan Task n
TIME
Hours, minutes, seconds and tenths
(output at start of scan)
(output at end of scan)
Done Task n
ENCODED
2 X X X X X X X X X (output at start of scan)
3 X X X X X X X X X (output at end of scan)
PRINTABLE ASCII CHARACTERS
␣ = space

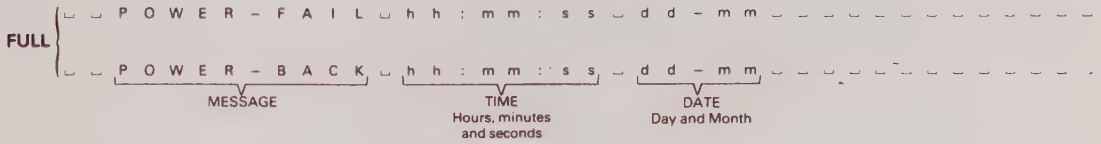
## 2.3 EVENT RESULT

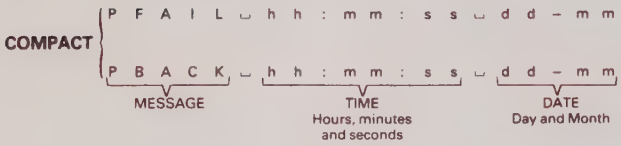



## 2.4 RUN AND HALT MARKERS



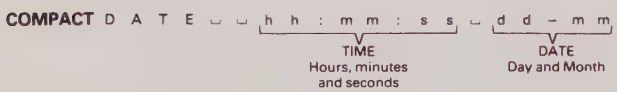
## 2.5 POWER FAIL AND POWER BACK MESSAGES



␣ = space

## 2.6 DATE



␣ = space

## 2.7 SPECIFIC FORMATTING

In addition to the general format information given on the preceeding pages, the following formatting details apply to the specified output devices only.

*Cartridge* Data is recorded in 1024 character blocks (the last 4 characters are always spaces). Partially filled blocks are filled with space characters, see section 4.

*Printer Terminal connected to PL1 on Controller (RS232)* Data is output in lines of up to 80 characters, terminated by C/R, or C/R LF and 4 nulls as determined by SW2 on Controller. Measurements are packed 2, 4 or 8 per line depending on Format. (The VDU mode suppresses LF if line contains exactly 80 characters).

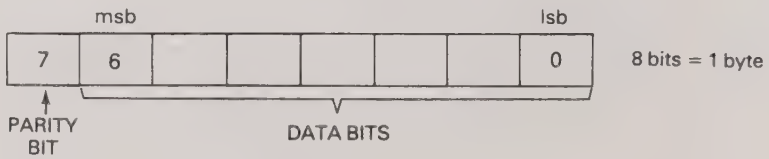
*GP-IB devices, talker only mode* Data is output in lines of up to 80 characters, terminated by C/R LF. Measurements are packed 2, 4 or 8 per line depending on Format. (See Chaper 11 for details.)

*RS422 port b (on Serial Interface)* Data is output in lines of up to 80 characters, termined by CR, or CR LF and 4 nulls. Measurements are packed 2, 4 or 8 per line depending on Format. (See chapter 10 for details.)

The terminating characters (eg C/R, LF etc.) are additional to the data characters.

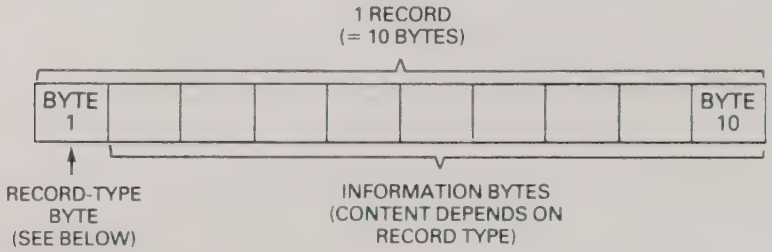
# 3. CODED BINARY FORMAT

Records output in Coded Format consist of ten-byte packets. Each byte, which is encoded as a printing ASCII character, consists of: 1 parity bit (even parity) and
7 data bits

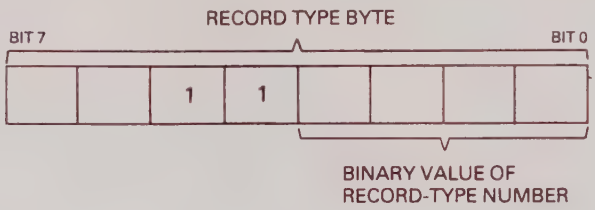



The most significant data bit (msb) is normally output as a '1'. This makes the byte into a printing ASCII character. When the binary value of the data alone exceeds 0111111, the msb assumes its true binary value.

All records consist of 10-bytes, although certain processed results consist of two records. The content varies according to the record type. The type, which depends on logger action at any given time, is indicated by the first byte in each record.



**Record-Type Byte** The four least-significant data bits indicate the type number.




| RECORD TYPE NO. | |
|---|---|
| 0 | Numerical Result: normal logged result |
| 1 | Numerical Result: monitored result |
| 2 | Start of Scan Marker |
| 3 | End of Scan Marker |
| 4 | Event Result |
| 5 | Run Marker |
| 6 | Halt Marker |
| 7 | Power Back |
| 8 | Power Fail |
| 9 | Date |

All Record types are detailed in the pages that follow. Each Record is shown in diagramatic form. The diagrams illustrate the overall content of the Records: the upper part of each diagram shows the numeric data which is generated within the logger, the lower part shows how the data is incorporated in the 10-byte Record which is output by the logger.

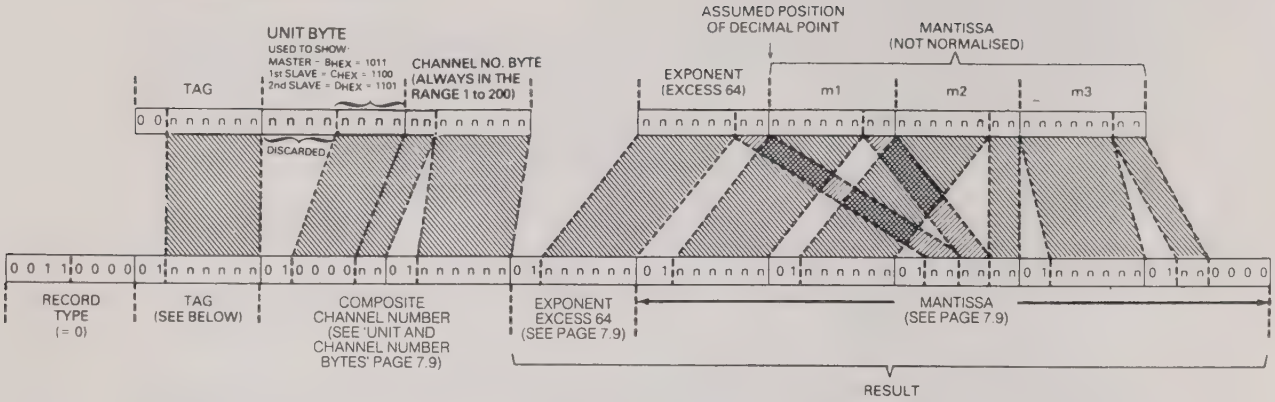
The various types of Coded format Records are listed below.

1. Numerical Result.
2. Start and End of Scan Markers
3. Event Result
4. Run and Halt Markers
5. Power Fail and Power Back
6. Data

The coded binary data record contains all the numeric information that the equivalent full and compact format records contain. However, for certain record types, the coded binary record will contain additional information not found in the full or compact record, ie:

a) The event record will contain the Task number
b) The Halt, Run, Power Fail, Power Back and Date records will all have the year included in the date.

## 3.1. NUMERICAL RESULT

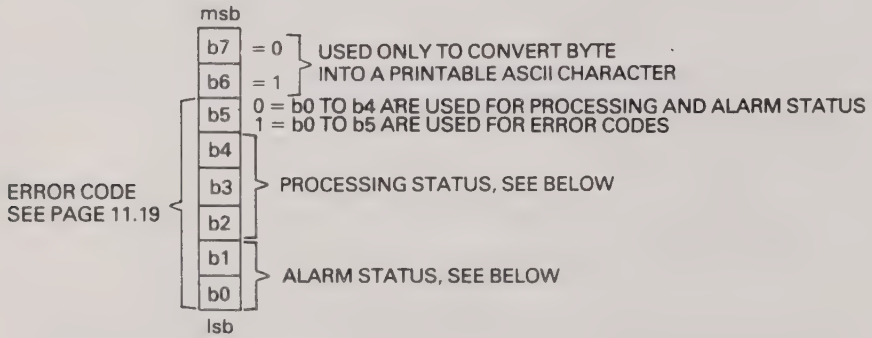



THE FORMAT FOR MONITORED RESULTS IS IDENTICAL, EXCEPT THAT THE RECORD TYPE (1ST BYTE) = 1

I.E. 0 0 1 1 : 0 0 0 1
RECORD TYPE (= 1)

**Tag Bytes** These are included in Numerical and Event results. A tag byte can convey processing and alarm status, or error codes.

The tag byte is eight bits wide, although the two most significant bits are used only to make the byte into a printable ASCII character. The significance of each bit is described below.




PROCESSING STATUS

| b4 | b3 | b2 | |
|---|---|---|---|
| 0 | 0 | 0 | No processing |
| 0 | 0 | 1 | Processed result, where original Channel reading is overwritten |
| 1 | 0 | 1 | Derived processed result, where original Channel reading still exists |

ALARM STATUS

| b1 | b0 | *(a) For all measurements except Event detection* | *(b) For Event detection only* |
|---|---|---|---|
| 0 | 0 | Non-alarm result | Invalid condition |
| 0 | 1 | < Low limit | Logic Hi-to-Lo transition |
| 1 | 0 | > High limit | Logic Lo-to-Hi transition |
| 1 | 1 | Invalid measurement condition or Match/Mismatch detected | Invalid measurement condition |

ERROR CODES

| b5 | b4 | b3 | b2 | b1 | b0 | ERROR NUMBER | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | Measurement overload |
| 1 | 1 | 1 | 1 | 1 | 0 | 2 | Measurement out of linearisation range |
| 1 | 1 | 1 | 1 | 0 | 1 | 3 | Invalid ambient temperature |
| 1 | 1 | 1 | 1 | 0 | 0 | 4 | Event total exceeds 65536 |
| 1 | 1 | 1 | 0 | 1 | 1 | 5 | Timeout in external-module measurement (>20 secs) |
| 1 | 1 | 1 | 0 | 1 | 0 | 6 | Strain gauge channel not initialised |
| 1 | 1 | 1 | 0 | 0 | 1 | 7 | Input connector not fitted |
| 1 | 1 | 1 | 0 | 0 | 0 | 8 | Result not yet accumulated (Counter Timer module) |
| 1 | 1 | 0 | 1 | 1 | 1 | 9 | Counter Timer Channel is off |

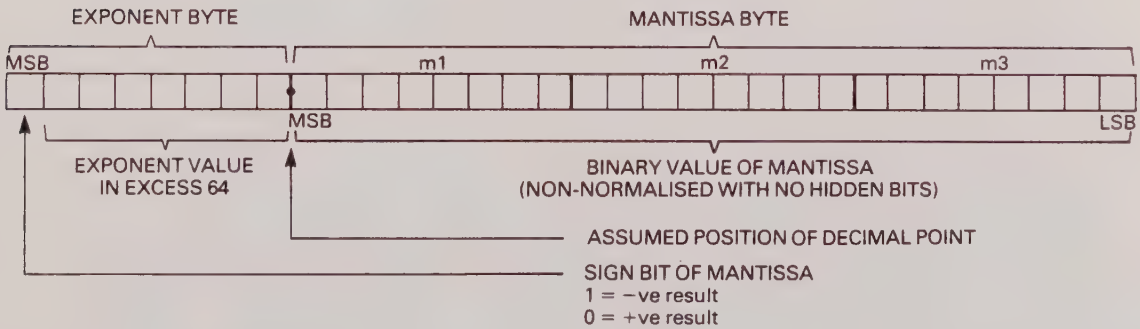
The error codes represent the 2's complement of the error number. If required, the error code may be extended by 2 bits to become an 8 bit 2's complement number, using external encoding. In this case bits 6 and 7 become '1', and the 8 bit number is read as 2's complement to obtain the error number.

**Unit and Channel Number Bytes** These produce a Composite Channel Number which indicates Unit (Master, 1st Slave, or 2nd Slave) and number within that unit (which is always in the range 1 to 200). However, in the case of the second record of a derived processed result, the Channel number is zero, indicating that the remainder of the record has no validity outside the logger.

Examples: Channel 005 (ie in the Master Unit) is shown as 1011 00000101
Channel 205 (ie in the 1st Slave Unit) is shown as 1100 00000101
Channel 405 (ie in the 2nd Slave Unit) is shown as 1101 00000101

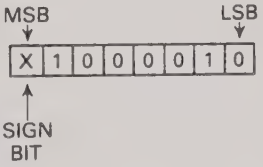
**Exponent and Mantissa**

Each result is expressed in standard floating point format, as an exponent and mantissa (m1, m2 and m3). The sign bit for the mantissa is contained in the Exponent byte (msb).



*Exponent Byte* The exponent byte contains the sign bit of the result (the msb) and the exponent value expressed as an Excess 64 number. To obtain the true exponent value, subtract 64.

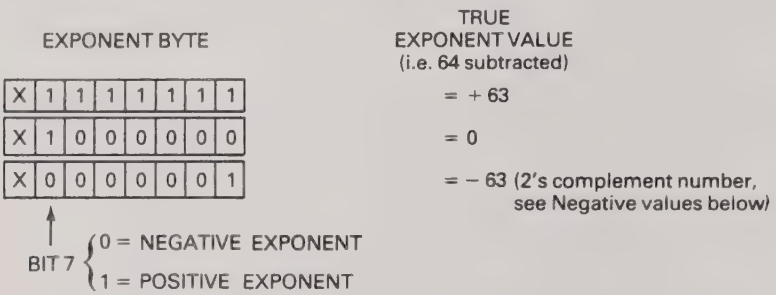
eg if the Exponent byte was




1. First take off the sign bit. The value then becomes: 1000010

2. Next, subtract 64 from the number; the value then becomes: 0000010
   Since $0000010_2 = 2$, the true value of the Exponent is 2.
   The exponent value (n) acts as a multiplier ($2^n$) on the mantissa.

The exponent value can be in the range +63 to −63. The bit patterns in the Exponent byte for true values of +63, 0 and −63 are shown below.


EXPONENT BYTE
TRUE EXPONENT VALUE (i.e. 64 subtracted)
X 1 1 1 1 1 1 1
= + 63
X 1 0 0 0 0 0 0
= 0
X 0 0 0 0 0 0 1
= − 63 (2's complement number, see Negative values below)
BIT 7
0 = NEGATIVE EXPONENT
1 = POSITIVE EXPONENT

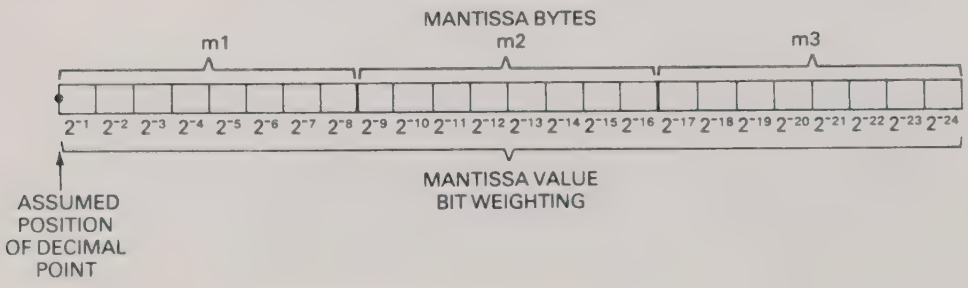
When the entire Exponent byte is zero, ie 0 0 0 0 0 0 0 0 , the number (exponent + mantissa) is assumed to be zero.

*Negative values* A negative exponent is expressed in 2's complement form. To convert the bit pattern of a negative exponent (ie bit 7 = 0); first subtract 64, then apply 2's complement. This gives the positive equivalen of the negative number.

eg An exponent of −7 is given by the following bit pattern: X 0 1 1 1 0 0 1

Subtracting 64 from 0111001 gives 1111001. Applying 2's complement to 1111001 gives 0000111. This equals 7, indicating that the exponent value is −7.

*Mantissa Bytes* The mantissa is a floating point, non-normalised value with no hidden bits. The bytes m1, m2 and m3 should be considered as one long byte which contains the mantissa value. The bit weighting and assumed position of the decimal point is shown below.


MANTISSA BYTES
m1
m2
m3
$2^{-1}$ $2^{-2}$ $2^{-3}$ $2^{-4}$ $2^{-5}$ $2^{-6}$ $2^{-7}$ $2^{-8}$ $2^{-9}$ $2^{-10}$ $2^{-11}$ $2^{-12}$ $2^{-13}$ $2^{-14}$ $2^{-15}$ $2^{-16}$ $2^{-17}$ $2^{-18}$ $2^{-19}$ $2^{-20}$ $2^{-21}$ $2^{-22}$ $2^{-23}$ $2^{-24}$
MANTISSA VALUE BIT WEIGHTING
ASSUMED POSITION OF DECIMAL POINT

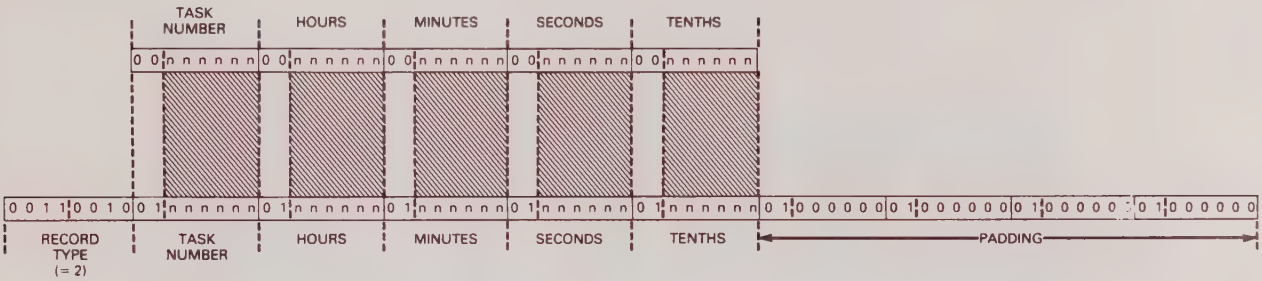
Mantissa polarity is given by the msb of the Exponent byte, 1 = −ve, 0 = +ve.

The effect of the exponent value (n) is to apply a multiplication factor of $2^n$ to the mantissa value. (This has the same effect as moving the decimal pont and all bit weightings; ie a positive exponent n will move the decimal point n places to the right, a negative exponent n will move the decimal point n places to the left).

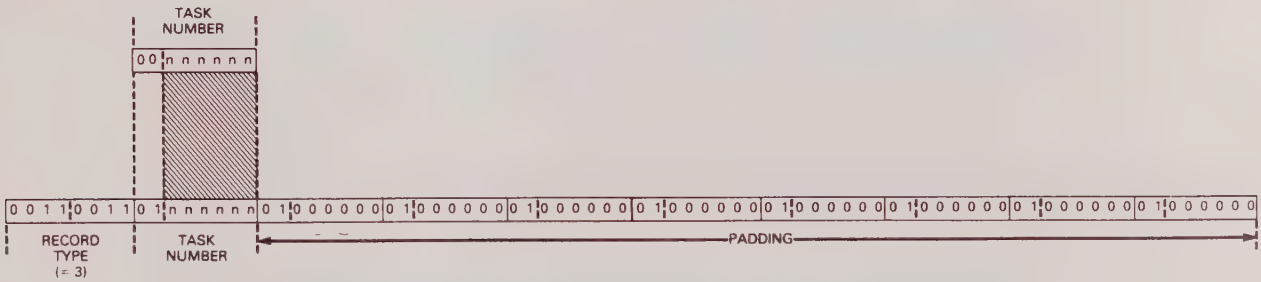
The Exponent byte and the Mantissa bytes are encoded into bytes 5 to 10 (inclusive) of the Numerical Result, as shown in the diagram on page

## 3.2 START AND END OF SCAN MARKERS

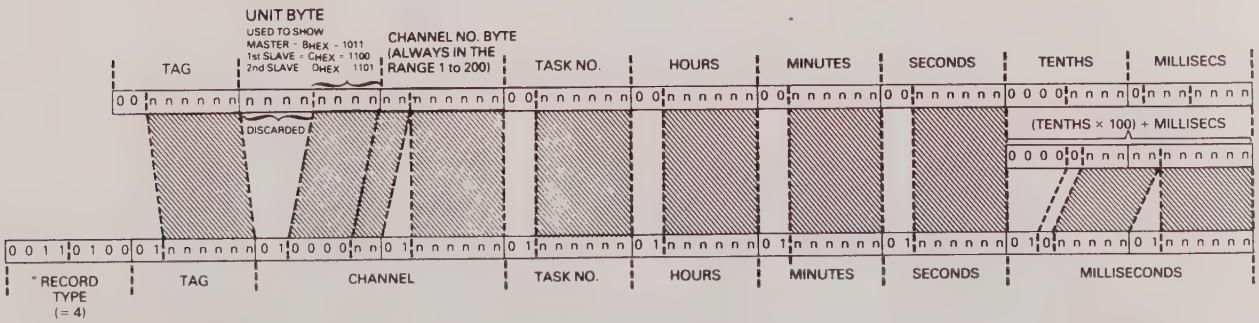
START OF SCAN




END OF SCAN

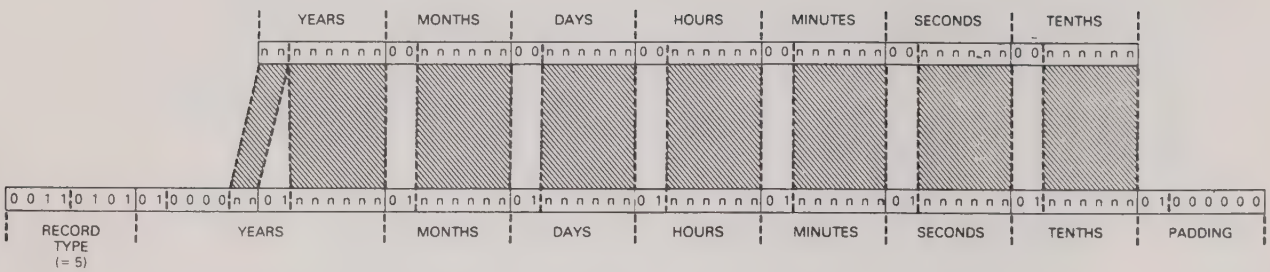



## 3.3 EVENT RESULT

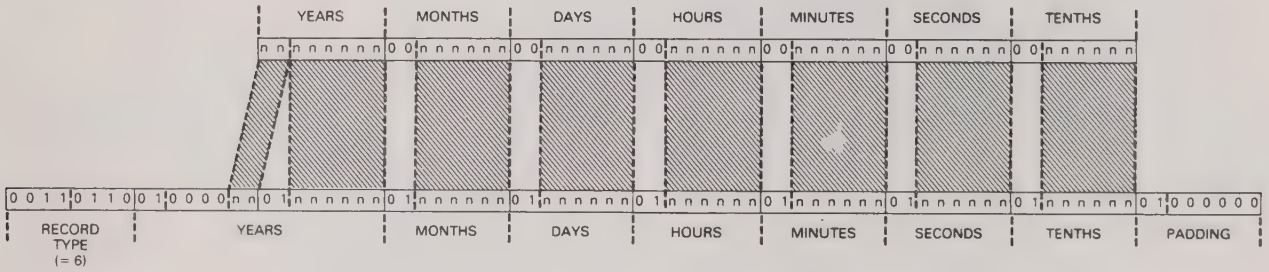


## 3.4 RUN AND HALT MARKERS

RUN MARKER

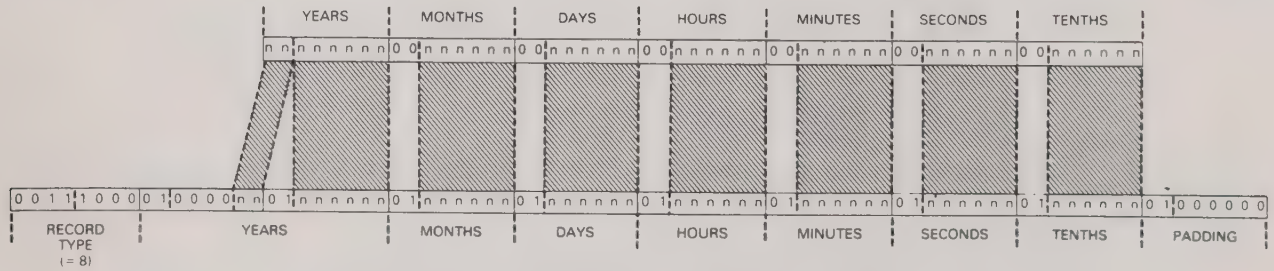



HALT MARKER

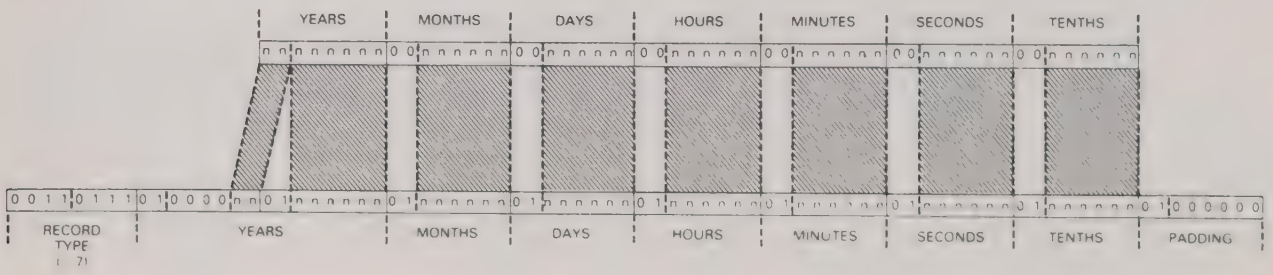



## 3.5 POWER FAIL AND POWER BACK

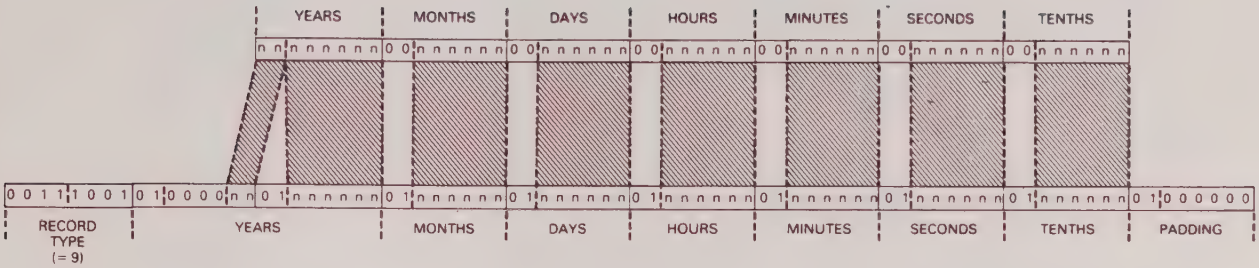
POWER FAIL




POWER BACK



### 3.6 DATE




## 4. RECORDING DATA ON CARTRIDGE (3530D ONLY)

The 3530D ORION Cartridge Recorder will read data from or write data to DC100 cartridge tape. The number of ORION measurements that can be written to tape will depend on the recording format of the measured data, coded, compact or full, and the number of slots reserved for ORION tape programs.

### 4.1 RECORDING MODES

The recording mode for measured data is selectable and can be Normal or Cyclic.

a) **Normal Mode** – blocks of data are written sequentially to track 1 until the tape early warning marker is detected by ORION. When detected ORION will finish writing the current block to track 1 and rewind to the start of track 2. Data is recorded on track 2 until it is full (end of tape) when recording will stop.

b) **Cyclic Mode** – data is recorded on tracks 1 and 2 as in normal mode except that when the end of track 2 is reached the tape is rewound to the start of track 1 and data that is already recorded is overwritten by subsequent data.

**Note:** Any program space that has been allocated at the start of track 1 is not overwritten, it is skipped over, recording does not recommence until the start of data marker has been reached.

### 4.2 RECORDING FORMATS

The measurement data will be recorded on a tape as 8-bit ASCII characters in one of the three recording formats; Full, Compact or Coded Binary.

**Compact and Full Format:** Cartridges recorded in Compact or Full format can be replayed in any suitable read station without the need for any decoding. One cartridge can accommodate a maximum of 25,000 measurements in Compact Format, or 12,500 measurements in Full Format.

**Coded binary format:** If ORION processing has not been used each 10 character ASCII string will contain numeric data in binary form, encoded into printable ASCII characters. Up to a maximum of 50,000 records can be accommodated in this form.

However, if processing has been applied, under certain conditions, each data record will be followed by a second 10 character record that will contain pointers to the ORION data areas and will be used by ORION when reconstituting previously recorded data. The amount of data that can be stored on tape may therefore be reduced by up to 50%.

Cartridges recorded in Coded binary format would normally be replayed in ORION, but can be replayed in other read stations if suitable decoding is applied to the data. However, if the tape contains processed data the records containing pointer information should be ignored as they contain no meaningful information for the user. The pointer record may be identified by a channel number of zero.

**Note:** When using ORION to decode a previously recorded tape containing processed data the program under which the tape was recorded must be reloaded into ORION. This program must be identical in every respect to the original program in order to obtain meaningful results.

## 4.3 DATA OUTPUT FORMAT

The data records are output to the tape packed in 1024 byte blocks, delimiting characters, ie carriage return and line feeds, are omitted from the output data string. The data record type may be identified however, by the first character of each string.

Full format – always a space character.
Compact – always an alpha character.
Coded binary – always a numeric character.

In addition to measured data the tape may also contain a copy of the complete ORION database. This is known as an ORION tape program, the format of a program, as saved on tape, is described in section 4.5.2.

## 4.4 ORION TAPE PROGRAM

An ORION tape program consists of a complete copy of the 3530D database. It will contain all the channel set up information, task definitions, processing programs, etc., that were in the ORION database when the save command was given, ie the program may contain information for up to eight tasks and associated processing programs.

Space must be reserved on tape for each saved program, a tape can contain a maximum of seven tape programs. The tape cartridge commands, detailed in chapter 5, section 3, are used to reserve space for, save and load programs.

The space reserved for each tape program will be 27 tape blocks. Space may be reserved for a maximum of seven tape programs and will always be at the start of track 1.

## 4.5 THE CARTRIDGE TAPE FORMAT

The cartridge is two track, both tracks may be used to record logged data and tape programs. The tape length is 150ft, up to 500k byte of information can be recorded giving a maximum capacity of between 12,500 and 50,000 measurements depending on recording format used.

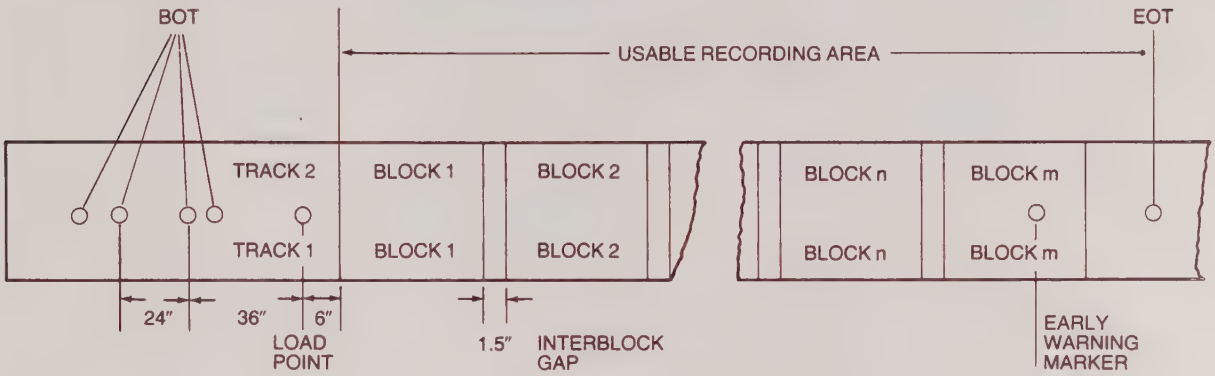



**Fig 7.1** *Physical tape Format*

The physical format of the tape is as follows:

| | |
|---|---|
| **First block** | – written 6 inches from load point. |
| **Block size** | – 1024 bytes. |
| **Interblock gap** | – approximately 1.5 inches of blank tape. |
| **End of Data marker** | – 45 inches of blank tape, from end of last block. |

**Note:** The end of data gap is only present if the end of tape has not been reached.

### 4.5.1 Data Only Tape

If no space has been reserved for Tape programs the whole of the tape may be used to record ORION data. The data will commence at the first block of track one, the data header block will **not** be present.

### 4.5.2 Program Tape

Space reserved for programs will always be at the start of track one.

**First Program Block**

The first block of each allocated program space will contain a header, the unused portion of the block will be padded with null characters. The header will be written to tape at the time the program space is allocated and will be of the form:

3530D▽▽DATA▽BASE▽n▽OF▽m

where n = program number
m = number of spaces reserved for programs on this tape.
▽ = space character

**Second Program Block**

When a program is written to a reserved space the second block will contain the program name in the form:

n:name

where n is the program number
name is the name given to the program, and is a character string of up to 40 characters.

**Data Header Block**
In order to identify the start of data the block immediately following the last block of reserved program space will contain a data header of the form:

MRL▽▽LOGGED▽DATA▽FILE

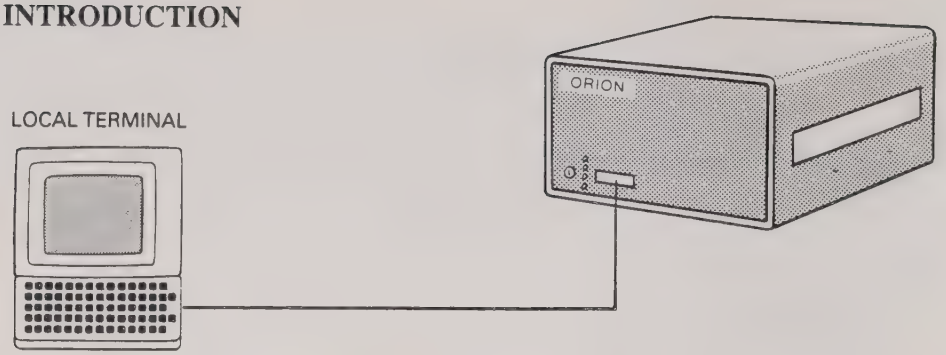
The remainder of the data header block will be padded with null characters. The data header will be written to tape when program space is allocated.

**Note:** The data header block is only present when space has been reserved for tape programs.

# Chapter 8
# Local Terminal Interface


| *Para. No.* | *Title* | *Page No.* |
|---|---|---|
| 1 | Introduction | 8.2 |
| 2 | Local Terminal Protocol | 8.2 |
| 3 | Protocol Summary | 8.3 |
| 4 | Connecting the Terminal | 8.4 |

# 1. INTRODUCTION




A VDU connected to the local terminal socket can be used to control 3530F ORION at distances or up to 50 feet. The local terminal is equivalent, in most respects, to the front panel of the 3530D, and will take full (local) control of ORION when no remote device is connected to one of the rear interface sockets. If a remote device is connected, the local terminal may lose full control of ORION but will retain a limited control, under which it can be used to monitor tasks or channels or take immediate measurements.

While the local terminal is being used to assert control of ORION it is not free to display task measurements or monitor results sent to it.

# 2. LOCAL TERMINAL PROTOCOL

A simple protocol is used to communicate between ORION and the local terminal.

**Gaining ORION's attention**
Before any commands can be set to ORION the terminal must first gain ORION's attention. This is done by typing ! at the keyboard and is referred to as 'establishing the control link'.

ORION acknowledges the request for attention by sending a status prompt (C or L), which is displayed at the terminal and the control link is established. The status prompt returned by ORION will indicate to the user whether he has full control or limited control of ORION.

**Controlling ORION**
With control established commands may be sent to ORION. The range of commands which can be used is dependent on the degree of control available at the terminal.

**Limited control (C prompt):** The following may be actioned when the terminal has limited control.

a. **Measure:** demand a single immediate measurement on any one channel.
b. **Query:** obtain the data from a specified part of the existing channel or task database.
c. **Monitor:** set up and/or start the monitoring of channels, tasks or headers.

Response to queries or measurement requests are displayed immediately at the terminal. However, the control link with ORION must be terminated before results from a monitor can be displayed at the local terminal. This is done by typing the # character at the terminal.

It may be possible for the terminal to regain full control of ORION. The command %8C/R typed at the terminal of a 3530F is equivalent to depression of the LOCAL button on a 3530D ORION. Providing LOCAL-LOCKOUT has not been asserted by a remote device, full control will be returned to the terminal and the prompt received from ORION will change to an L.

**Full control (L prompt):** All commands available under limited control, may be used. In addition, under full control, the terminal can be used to program ORION, ie set up and modify channel and task databases, run, halt, etc. Chapters 3 to 6 contain detailed information on the command language.

As with limited control, the control link with ORION must be terminated before results, other than query or immediate measurements, can be displayed at the terminal.

**Relinquishing control**
A single character # typed at the terminal keyboard will indicate to ORION that the user no longer wishes to send commands to it. On receipt ORION will immediately terminate the control link with the terminal. If the terminal has been designated to receive either task or measurement results, or monitor data these will be displayed as soon as they become available.

**Note:** Typing a # at the terminal will always terminate the control link, even if typed in the middle of a command string.

## 3. PROTOCOL SUMMARY

| TERMINAL INPUT | | ORION RESPONSE | |
|---|---|---|---|
| **Code** | **Meaning** | **Code** | **Meaning** |
| ! | Attention, request control | L<br>C | You have local control<br>Another device has control |
| %8C/R | Request control if ! response was C | L<br>E113 C | You have local control<br>Another device has local lock-out |
| # | Relinquish control | None | Any data awaiting display will be output |

**L response** The terminal has Full Control and may be used to:

a. **Program:** set-up and modify channel and task data bases, cause ORION to Run, Halt, etc. Modification may not be made while ORION is running.

b. **Measure:** demand a single immediate measurement of any one channel. This result will be displayed without further action.

c. **Monitor:** requests a monitor of specified channels or tasks.

d. **Query:** display a specified part of existing channel or task data bases. Queried information is displayed immediately.

**C response** The terminal has limited Control. In this event Programming (a. above) is not available; an attempt to do it will be ignored, and will result in the error response E110C. Immediate measurement, monitor and query commands may be used.

**Displaying results** Having relinquished control (#), the local terminal is available to display the results from the programs that are running in ORION. This it will do if specified as the output device.

**Error messages**
E113 results from %8 if ORION in local lock-out by another device.
E110 results from attempting to program when Full Control has not been established.

## 4. CONNECTING THE TERMINAL

Fixed data rate: 9.6k baud.

Output line termination Protocol: C/R LF, or C/R only if line length is exactly 80 characters.

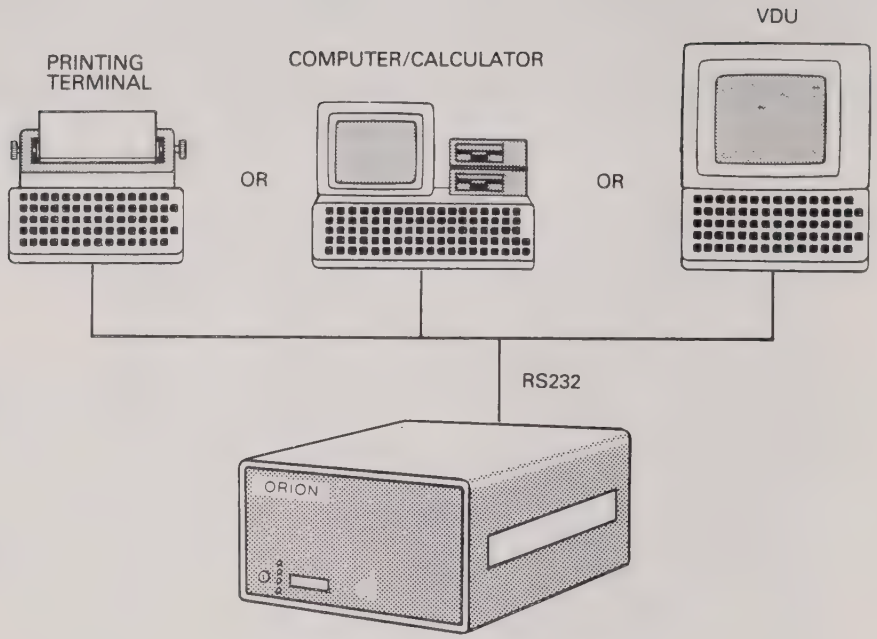
**Pin connection details**

| Pin No. | Signal |
|---|---|
| 2 | TX Data |
| 3 | RX Data |
| 4 | RTS (Request to send) |
| 5 | CTS (Clear to send) optional |
| 7 | Signal Ground |
| 20 | DTR (Data Terminal Ready) |

# Chapter 9
# Standard RS232 Interface


| *Para. No.* | *Title* | *Page No.* |
|---|---|---|
| 1 | Introduction | 9.2 |
| 2 | Operating with a computer using the Standard RS232 Interface | 9.3 |
| 2.1 | Result Transfer mode | 9.3 |
| 2.2 | Control Mode | 9.4 |
| 2.3 | Communication using simple Protocol | 9.7 |
| 3 | Operating with a VDU or Printing Terminal | 9.11 |
| 3.1 | Result Transfer Mode | 9.11 |
| 3.2 | Control Mode | 9.11 |
| 3.3 | Relinquishing control | 9.11 |
| 4 | Output Only protocol | 9.11 |
| 5 | Connecting to Standard RS232 Port | 9.12 |

# 1. INTRODUCTION



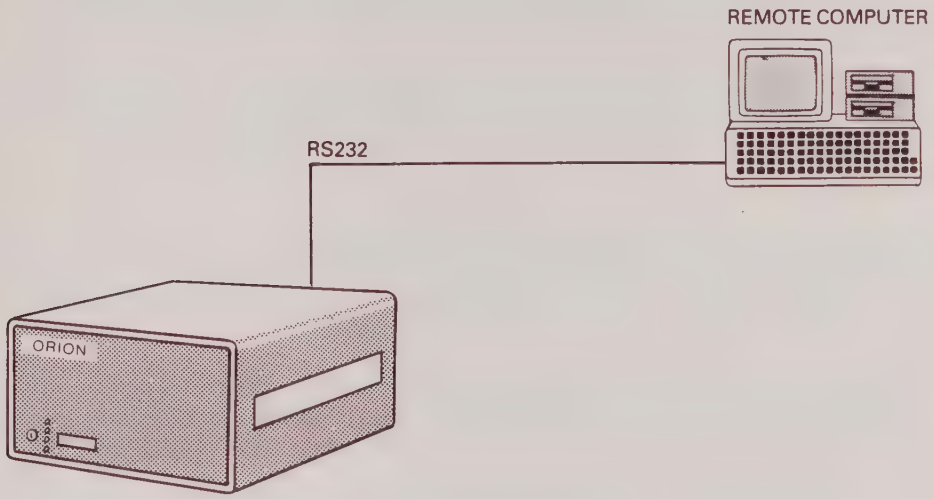

The RS232 socket on the controller board is suitable for connection to a device capable of remotely controlling ORION, such as a computer, calculator or VDU. Alternatively, it may be used to connect an output only recording device such as a printer or punch. Data rates and output protocol are switch selectable.

**Computer/calculator:** ORION can be controlled using a computer or calculator such as Apple with California computer serial I/O or an HP calculator with minimal interface. Hardware handshake is not required, although one character hardware buffering is (ie ACIA). Output lines are terminated by C/R only. Section 2 describes the communications protocol adopted.

**VDU or Printing Terminal:** any suitable VDU may be connected to control ORION and/or display results. Output lines are terminated by C/R LF, or C/R only if the data blocks are 80 characters long. Alternatively a printer with keyboard, eg Silent 700 ASR or DEC LA38 teletype, can be used if hard copy of measurement results is required. Output lines are terminated by C/R LF, and 4 nulls. The communications protocol used is described in Section 3.

**Output only device:** a suitable punch or cartridge recorder may be connected to record measurement data. Output lines are terminated by C/R only. Alternatively a low speed printer may be connected to produce hard copy of results. Output lines are terminated by C/R LF, and 4 nulls. The communication protocol is described in Section 4.

## 2. OPERATING WITH A COMPUTER USING THE STANDARD RS232 INTERFACE




Two distinct modes of operation are possible:

1. Transfer of results from ORION to the computer.
2. Control of ORION by the computer to set up data bases, run or halt ORION or to ask ORION for certain information.

These two modes are exclusive, in that if the computer is controlling ORION results cannot be transferred.

### 2.1 RESULT TRANSFER MODE

When tasks within ORION are run, results for each channel are fed into a buffer store. If the standard RS232 is designated as the output port ORION will offer to transfer as soon as the first result is in the buffer. This is does by sending **O** to the computer, which replies with **?** and the result is then output. Successive interchanges of O and ? enable further results to be passed to the computer.

It will be seen that the results are not solicited by the computer; they may be offered at any time. The computer must therefore include at least a single character buffer it observes from time to time to see if ORION has sent an O. This must be done sufficiently frequently to ensure that the buffer (600 results) does not fill and thereby prevent ORION continuing with its tasks.

To receive results it is not necessary to set-up or otherwise control ORION from the computer; this may have been done at the ORION front panel (3530D) or from a local terminal (3530F). In this case the computer acts as an output peripheral which receives results by the O and ? routine. However if the computer has been used to control, thereby occupying the communication link, the computer must release the link so that transfer of results may occur; this it will do by sending #.

## 2.2 CONTROL MODE

When the computer has gained full control it will be able to set up and modify the ORION database. In addition it may command ORION to perform other duties such as monitor channels or tasks, or perform an immediate measurement on a channel.

In order to exert control the computer must occupy the communications link. This it does by sending ORION the single character.

!

The link will then no longer be available for the transfer of results.

**N.B.:** The request to establish the communications link (!) will not be successful if there is a prompt for data available outstanding.

ie If ORION has sent O, data ready, for which it has not received a request for transfer,?, it will ignore the request to establish the link,!, until the O has been acknowledged.

In reply to the computers request to establish the link the ORION will respond with one of four status prompts. Th_ ,e prompts define the level of control that the computer has gained over ORION:

**N** ORION does not recognise the computer as a remote device that wishes to control it. The computer must identify itself to the computer by sending the %1 C/R command in order to gain full control.

**L** ORION recognises the computer as a potential controller, however, at the moment ORION is in 'LOCAL' control. The computer should send the %4 or %6 command to gain full control.

**C** The computer has full control of ORION.

**P** The ORION recognises the computer as a potential controller, but at the moment the computer cannot attain full control of ORION because a high priority device is already in control.

**Note:** Of the remote interface at the rear of the ORION the standard RS232 interface has the lowest priority.

### 2.2.1 Gaining Control of ORION

When the computer is connected to the ORION and switched on ORION will normally be in the LOCAL state. The ORION response after establishing the link will be the N status prompt, in order to gain full control the computer must send the command.

**%1**C/R This identifies the computer to ORION as a potential controller. The expected response would be a L or C prompt, however, if a higher priority device has already taken control the prompt will be P.

In response to the L or C prompt the computer may send either of the following commands to gain full control.

**%4** C/R Request full remote control of ORION if response was L. ORION can be expected to reply with C.

**%6** C/R Put ORION into local lockout and request full control, if the computer does not already have it, the expected response from ORION is C. The lockout prevents interference with the database either from the front panel of 3530D or local terminal connected to a 3530F.

**Note:** Once the communications link has been established the computer will always have 'limited' control over the ORION. However, 'full' control is only established when the ORION replies with the C status prompt.

### 2.2.2 Limited Control

When the computer has occupied the communication link by sending ! it has limited control which may be exercised whichever response (N, L, C or P) is received from ORION. In limited control the following may be actioned:

a. **Measure:** demand a single immediate measurement of any one channel.
b. **Query:** obtain the data from a specified part of existing channel or task data bases.

For both Measure and Query, ORION checks the validity of the command and sends E or R.

**E** means command in error. ORION ready to send error information.

**R** means the command has been actioned and ORION is ready to send data.

On receipt of E or R, the computer sends ? and ORION then outputs the data as a string of up to 80 characters terminated by C/R. When the data is an error code it will be of the form Exxx C/R, where xxx identifies the kind of error.

Following the data string the ORION will output the single status prompt of N,L,C or P.

### 2.2.3 Full Control

When the response from ORION is C the computer has full control. In addition to Measure and Query that are available with limited control, the computer can also:

a. **Program:** set up and modify channel and task data bases, run, halt etc.
b. **Monitor:** request a monitor of specific channels or tasks.
c. **List:** from 3530D only. ORION sends to the computer the information held in specified channel and task data bases. To use this the computer would have to output the list to an associated VDU or printer, or store the list for subsequent output.
d. **Read Cartridge:** from 3530D only. Programs and/or data recorded on the cartridge within ORION may be read to the computer, see Chapter 7 for details of recording formats and copy facility.

In the case of Monitor, List and Read Cartridge the data are made available only after the computer has sent a #, and are then transferred using the O and ? routine.

### 2.2.4 Communicating with ORION

When the link has been established and the computer has received an ORION status prompt it should reply by sending either a:

**'command string'** C/R to communicate with ORION

or alternatively, if the link between ORION and the computer is to be terminated the character:

# to terminate control mode.

The type of commands that may be sent to ORION will depend on whether the computer has 'full' or only 'limited' control of ORION .

When the reply is a 'command string' ORION will check its validity before responding with the prompt:

**E, R, N, C, P or L**

where: E = command input was in error, ORION is ready to output error information.

R = command accepted, ORION is ready to output data in response to the command.

N,C,L or P = command accepted, ORION ready to accept the next command.

If the E or R prompt was received the computer must respond with a question mark:

?

whereupon ORION will send up to 80 characters of resonse data followed by a carriage return (C/R). If the response was an error message it will be in the form:

**E nnn** C/R

where: nnn is the numerical error code.

Following the response data the ORION will output the single status prompt character, N,C,L or P as appropriate.

The ORION is now ready to receive further command data or a hash (#) to terminate control mode input.

### 2.2.5 Relinquishing Control

The computer may relinquish control of ORION in one of three ways:

**#** By sending # the computer releases the communications link and permits ORION to offer (O) results of measurement or other information that the computer may have requested. The # does not destroy the control status already set up by the computer. This status is regained if ! is sent.

**%5** C/R This tells ORION to release local lock-out and assume local status. With 3530D the front panel then has control; with 3530F the local terminal is able to gain control. The computer retains the RS232 link and may regain full control by sending %4 or %6. Results cannot be output unless # is sent.

**%3** C/R The computer relinquishes the status of controller, but if ORION is in local lock-out this command does not remove it. The state reverts to N. This permits another remote device to take control at some future time; meanwhile any intervention at the front panel or local terminal is precluded. Results cannot be output unless # is sent because the computer still occupies the RS232 link. Full control may be regained by sending %1, followed by %4 or %6 as appropriate.

## 2.3 COMMUNICATION USING SIMPLE PROTOCOL

The flow diagrams in figures 9.1 to 9.3 are examples of the communication protocol required to communicate with an ORION when using an interrupt driven computer, such as the HP-85.

**Figure 9.1** shows a main control loop in which the program will sit whilst waiting to send a command to, or receive data from, ORION.

**Figure 9.2** shows two keyboard interrupt routines, one that sends a character to ORION in order to gain control of the communications link and a second that sends a character to ORION to terminate the link.

**Figure 9.3** shows an ORION interrupt service routine that checks the validity of the prompt character received from ORION and takes the appropriate action.

For computers, such as the APPLE and PET, that do not support interrupts it will be necessary to monitor the input buffer and keyboard buffer for input. The monitoring should take place at frequent intervals so that prompts are not missd. The interval between receiving the last character of ORION data, and the next prompt being sent, is dependent on the amount of processing being carried out by ORION, and transmission baud rate selected.

**Note:** In theory the prompt could be sent immediately following the stop bit of the final data character, however in practice there is normally a short delay.

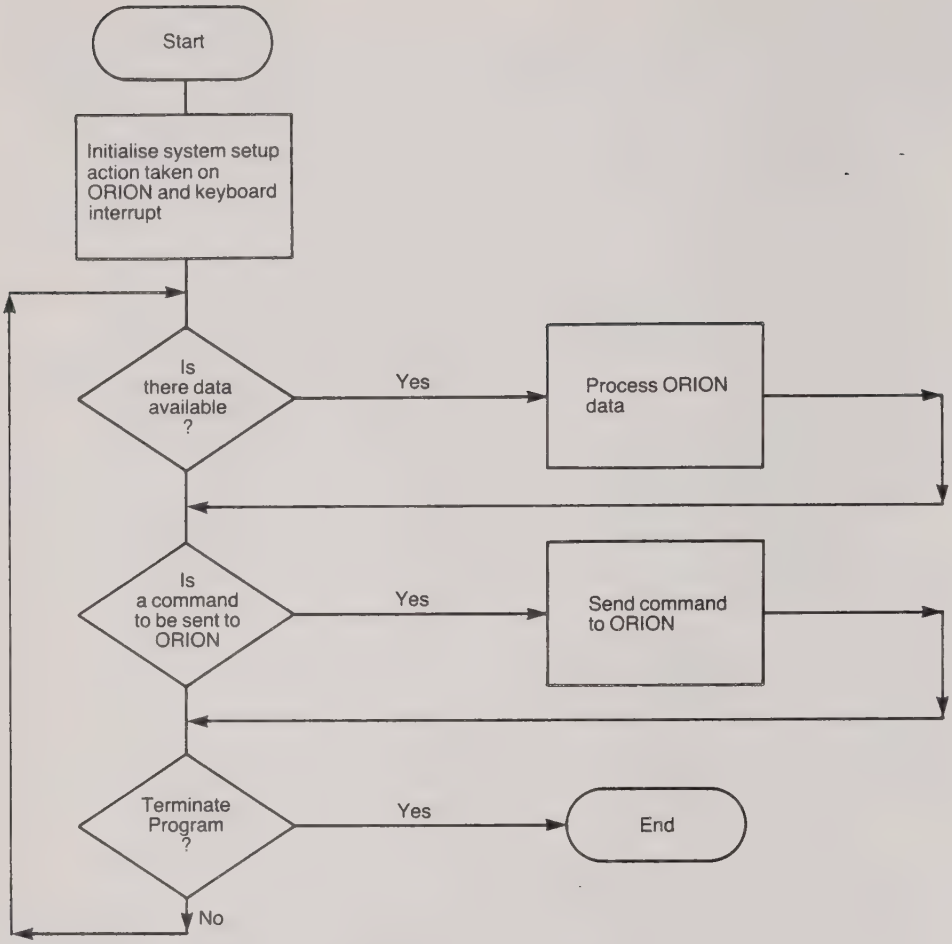



**Fig 9.1** *Main Control Loop*

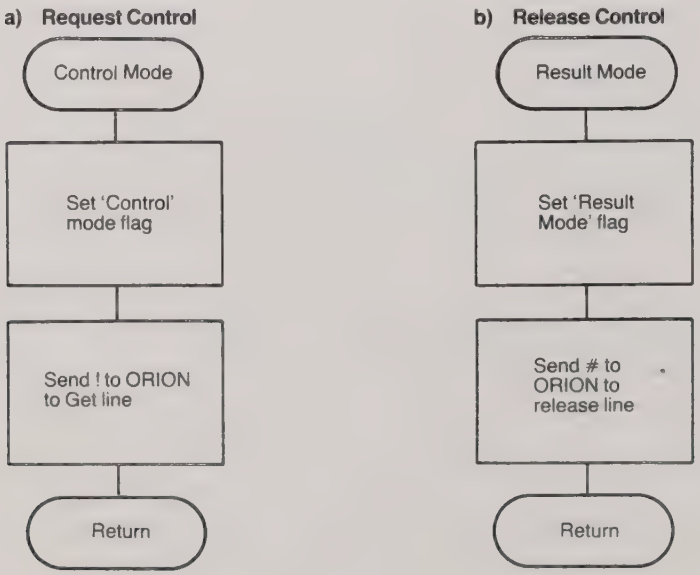



**Fig 9.2** *Keyboard Interrupt routines*

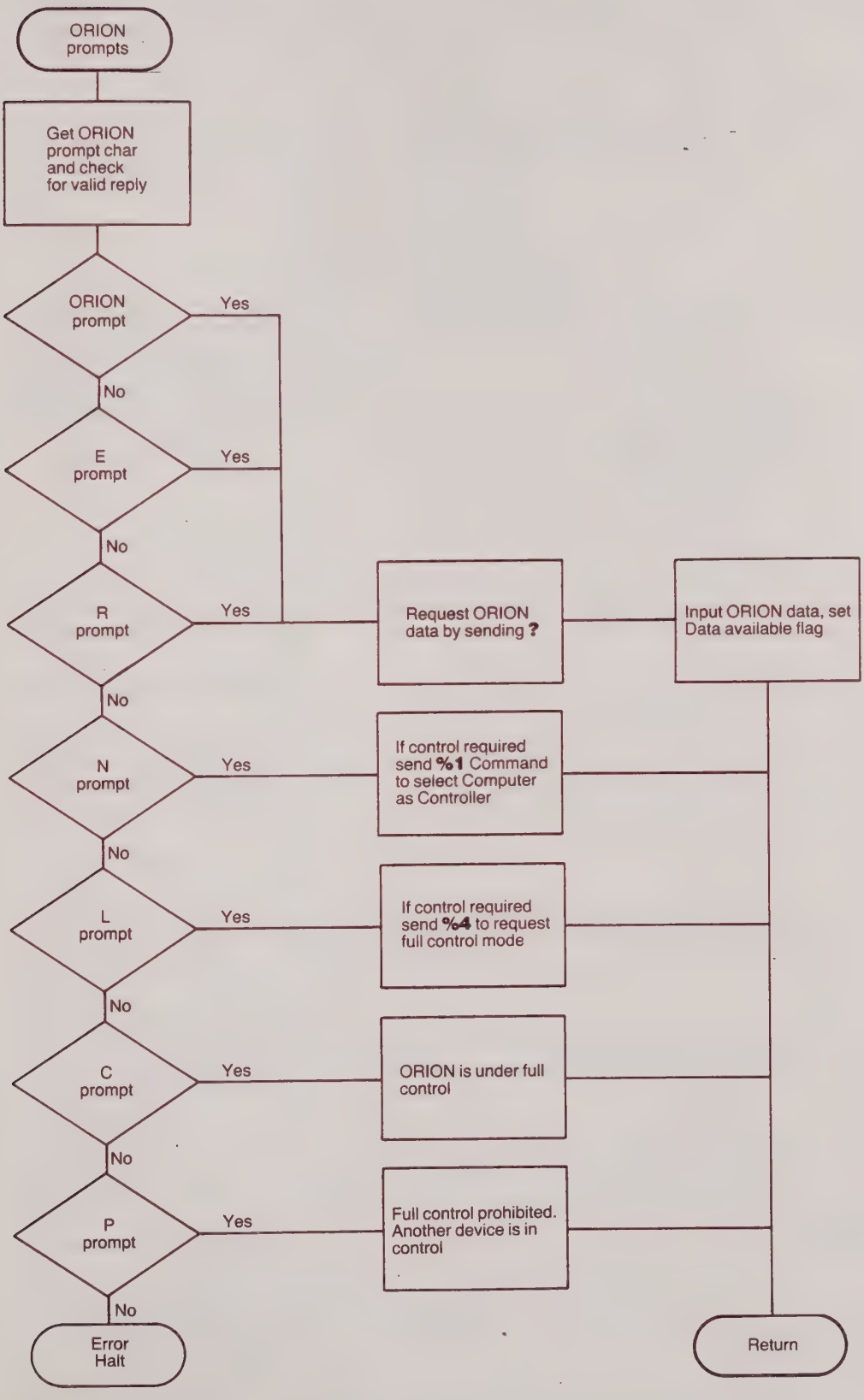
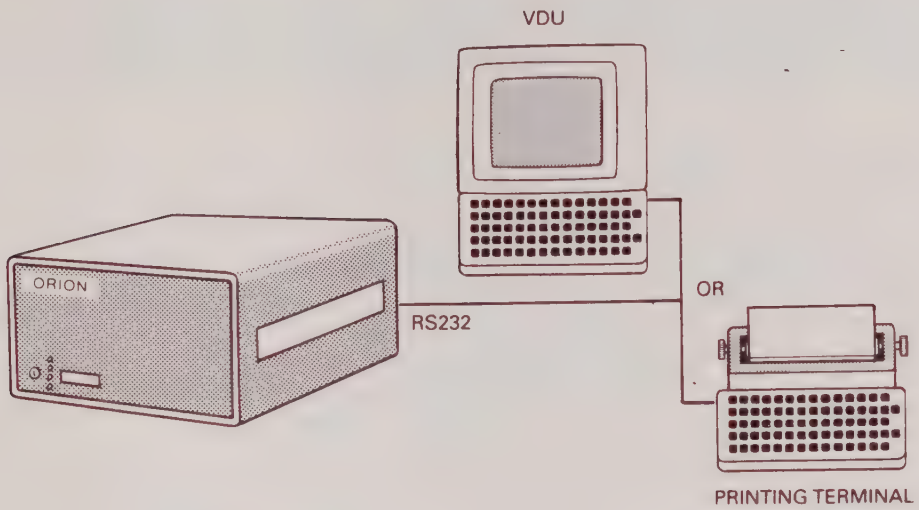



**Fig 9.3** *ORION Interrupt Service Routine*

| COMPUTER INPUT | | ORION RESPONSE | |
|---|---|---|---|
| Code | Meaning | Code | Meaning |
| ! | Attention. Computer occupies the communications link. | N<br><br>L<br><br><br>C<br>P | You (computer) have not identified yourself as a controller.<br>I (ORION) am in local, but you are recognised as remote controller.<br>You have full control.<br>High priority device has control. |
| %1 C/R | If response was N.<br>Computer identifies itself as a potential controller. | C<br>L<br><br>E | You have full control.<br>I am in local, but you are recognised as remote controller.<br>Error E113 the command has been input out of sequence. |
| %4 C/R | If response was L, request full control. | C<br>E | You have full control.<br>Error E113 the command has been input out of sequence. |
| %6 C/R | Assert local lock-out and gain full control. | C<br>E | You have full control.<br>Error E113 the command has been input out of sequence. |
| %5 C/R | Release local lock-out and go to local. | L<br>E | I am in local.<br>Error E113 the command has been input out of sequence. |
| %3 C/R | I relinquish controller status, but not local lock-out. | N<br><br>E | You are no longer recognised as a controller.<br>Error E113 the command has been input out of sequence. |
| Command String C/R | Send a command or series of commands to ORION. | C<br>P<br><br>E | Commands accepted.<br>A higher priority device has taken control.<br>Error E113 the command has been input out of sequence. |
| Measure Command C/R | Request an immediate measurement. | R<br>E | Result ready for output.<br>Error in command input. |
| Command ? C/R | Query command. | R<br>E | Query data ready.<br>Error in command input. |
| ? | Request Error number, immediate measurement data or query data. | "data" C/R<br><br>N,L,C or P | Error number in for nnn or upto 80 char. of query or measurement data,<br>followed by the status prompt. |
| # | Release the link to permit transfer of results. | | |
| **RESULT TRANSFER MODE** | | | |
| | | O | Data ready to send. |
| ? | Request transfer of data. | data C/R<br>O | Up to 80 characters of result data, followed by the O prompt when ready to send further data. |

**Fig 9.4** *Summary of Commands and Response*

# 3. OPERATING WITH A VDU OR PRINTING TERMINAL




The protocol used with a VDU or printer with keyboard is similar to that described for a computer in the previous Section. The differences in protocol are outline in the paragraphs which follow.

## 3.1 RESULT TRANSFER MODE

Data is output to the terminal immediately it becomes available. However data cannot be displayed or printed while the terminal is occupying the communications link. Any data sent while the communications link is established will be stored in the buffer (600 results). To receive data the link should be terminated by typing a # at the terminal.

## 3.2 CONTROL MODE

As described in Section 2.2 the communications link is established by typing an ! at the terminal and the prompts returned by ORION are N, L, C or P. The response input at the terminal to gain full control of ORION is %1, %4 or %6.

**Limited Control**
Measure and Query command may be used under limited control as described in Section 2.2.2 except that response data is output to the terminal immediately. The E and R and ? response and commands are not required under this protocol.

**Full control**
All commands available under limited control may be used. In addition Program, List, Monitor and Cartridge commands may be used as described in Section 2.2.3.

## 3.3 RELINQUISHING CONTROL

The communications link is terminated by typing a # at the terminal. The %3 and %5 commands are used to relinquish remote status without terminating the control link as described in Section 2.2.5.

**Note:** Typing # at any point in a command line will result in the link being terminated immediately.

## 4. OUTPUT ONLY PROTOCOL

The standard RS232 port can be used to connect ORION to an external recording device such as a low speed printer, cassette recorder or paper tape punch. ORION will send data to the output port as it becomes available, if the device is not ready to receive data the data will be lost.

## 5. CONNECTING TO STANDARD RS232 PORT

When connecting a terminal device to the standard RS232 socket on the ORION controller PCB, the correct switch setting for Baud rate and protocol must be selected. Details of switch setting and the connector pin (PL1), are given below.

SW1/SW2 switch setting for Baud rate

| Switch Contacts | Function | Switch *Off | Position On |
|---|---|---|---|
| SW1/1 | Select Baud Rate | | 75/110 |
| SW1/2 | | | 150B |
| SW1/3 | | | 300B |
| SW1/4 | | | 600B |
| SW1/5 | | | 1.2kB |
| SW1/6 | | | 2.4kB |
| SW1/7 | | | 4.8kB |
| SW1/8 | | | 9.6kB |
| SW2/1 | Select Baud Rate | 75B | 110B |

*All contacts should be set off except for the one for the Baud Rate selected.

SW2 switch settings for output protocol

| Device | Controller Board Switch Setting (SW2) S3 | S2 | S1 | Output Line Termination | Orion Prompts |
|---|---|---|---|---|---|
| COMPUTER CALCULATOR | ON | ON | OFF | C/R | L N C P R E O |
| VDU | OFF | ON | OFF | C/F LF or C/R only if 80 char. line | L N C P |
| PRINTER | OFF | OFF | ON | C/R LF 4 nulls | none – o/p only device |
| OUTPUT ONLY PRINTER WITH KEYBOARD | OFF | OFF | ON | C/R LF 4 nulls | L N C P |
| TAPE PUNCH | ON | OFF | OFF | C/R | none – o/p only device |

PL1 – Pin details

| Pin No. | Signal |
|---|---|
| 2 | TX DATA |
| 3 | RX DATA |
| 4 | RTS (REQUEST TO SEND) |
| 5 | CTS (CLEAR TO SEND) |
| 7 | SIGNAL GROUND |
| 20 | DTR (DATA TERMINAL READY) |

# Chapter 10
# Fast Serial Interface

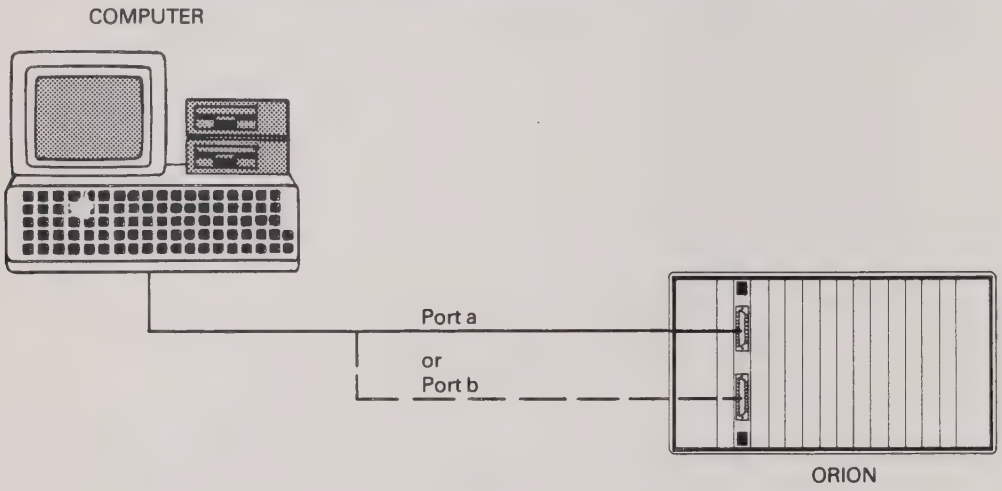

| *Para. No.* | *Title* | *Page. No.* |
|---|---|---|
| 1 | Introduction | 10.2 |
| 2 | RS422/232 Operation with BS4505 Protocol | 10.2 |
| 3 | Other RS232 Protocol Available | 10.7 |
| 4 | Serial Interface Connection Details | 10.8 |
| 4.1 | Cable for RS422 Operation | 10.9 |

## 1. INTRODUCTION

The serial interface board 35304B has two output ports which can be set independently for either RS422 or RS232 operation. BS4505 has been fully implemented for both interface standards. In addition limited RS232 protocol is available to control external recording devices such as a Penny and Giles recorder, printer or tape punch.

If either of the ports are being used to control ORION a request by either port for Remote status will take precedence over the standard RS232 port on the controller board. In addition port B is of higher priority than port A.

## 2. RS422/232 OPERATION WITH BS4505 PROTOCOL




The communications protocol as defined in British Standard 4505 is primarily designed for use with equipment capable of being driven at high baud rates and/or over long distances via a RS422 balanced voltage connection, eg a computer. However, the system is equally suitable for less distant RS232 operation by changing the connecting cables and selecting the appropriate data rate and operating mode.

The protocol, as defined in BS4505, involves the communication between two devices. ORION has been hardware configured so that it is always the lower priority device. Therefore if a request for control is made by the other, eg a computer, that device will always gain control of the line.

**ORION BS4505 Protocol**

The device wishing to transmit data must first gain control of the line. This is done by sending an enquiry (ENQ) to the other device. Not until a positive acknowledgement, (ACK), is returned by the receiving device will the device wishing to transmit have gained control of the line. Once a device is in control it may send data in the form of a data block.

The format of the data block is defined as follows:

| | | | | | |
|---|---|---|---|---|---|
| SOH | HEADER | STX | DATA ..... | EXT or ETB | BCC |

where: SOH — Start of header marker.
HEADER — Characters indicating data type (see table 10.1).
STX — Start of data marker.
DATA — Data block – max. of 80 characters for ORION protocol, excluding terminating characters (C/R or C/R LF).
ETB — End of data block marker – when another block of data of same type is to follow.
ETX — End of text marker – when this is the last block of data of this type and the line is to return to the neutral state.
BCC — Block check character.

**Note:** All characters are: asynchronous, 7 data bits + 1 parity bit, even parity.

The receiving device will check the block of data for validity, and if valid will acknowledge its receipt with an ACK, allowing a further block to be transmitted if available. If the data received was corrupt a negative acknowledgement, NAK, is returned to the transmitting device and the last block sent is re-transmitted.

The data block transmitted after the link has been established will contain a start of header marker, SOH, followed by the header characters. The header consists of two ASCII characters which identify the type of data being transmitted in this block. Table 10.1 gives the value and meaning of the header markers used by ORION.

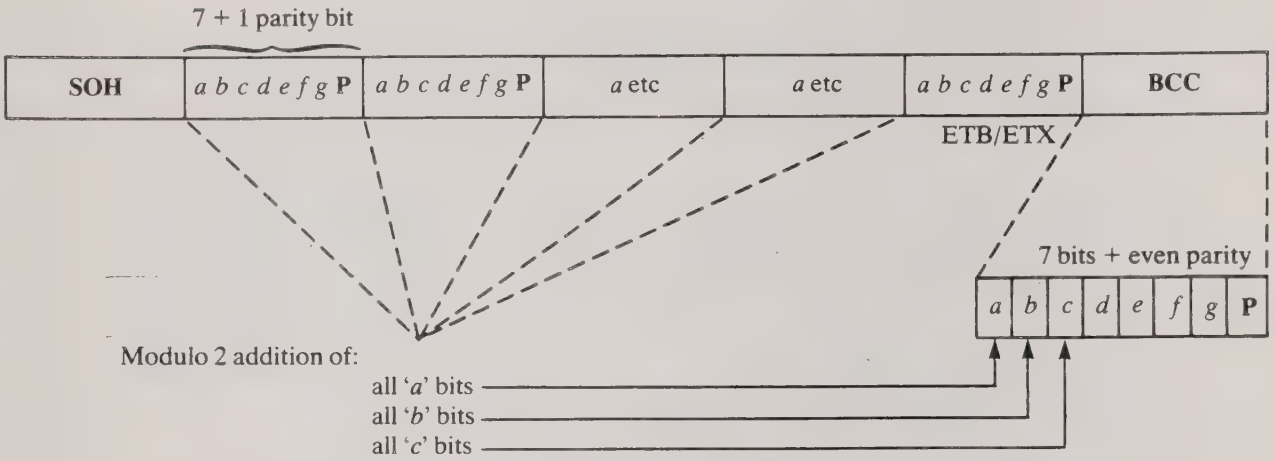
**Table 10.1** Header Characters

| 1st Byte (ASCII) | 2nd Byte (ASCII) | Description of Data |
|---|---|---|
| 0 | | Monitor data stream |
| | 0 | Monitor task result |
| | 1 | Monitor channel result |
| | 2 | Monitor start marker |
| | 3 | Monitor end marker |
| | 4 | Monitor event |
| 1 | | Logged data stream |
| | 0 | Ordinary logged result |
| | 2 | Start Scan marker |
| | 3 | End of Scan marker |
| | 4 | Event Result |
| | 5 | Run Marker |
| | 6 | Halt Marker |
| | 9 | Date Message |
| 2 | | Copied data from tape i.e. previously logged data |
| | 0 | All data |
| 3 | | Messages |
| | 0 | Ordinary messages |
| | 7 | Power Fail Message |
| | 8 | Power Back Message |
| 4 | | Listed data: ASCII format, data produced by LIST command |
| | 0 | All data |
| 5 | | Command response: ORION replies to commands |
| | 0 | Non error responses |
| | 1 | Error responses |
| 6 | | Commands: command data sent to ORION |
| | 0 | All commands |

The final character of the transmitted block is a check character, the purpose of which is to check that all characters in the block were received uncorrupted. This block check character (BCC) is 7 bits plus a parity bit (even parity), where each of the first 7 bits are the modulo 2 binary sum of every element in the corresponding bit column, 1 to 7, of successive characters of the transmitted block. The BCC is generated by performing an exclusive OR on all characters transmitted after the SOH, up to and including the ETB or ETX character.

**Note:** The **SOH** character is **not** included in the check.

The method of error detection is often referred to as 'the longitudinal parity method'.


7 + 1 parity bit
SOH
a b c d e f g P
a b c d e f g P
a etc
a etc
a b c d e f g P
BCC
ETB/ETX
7 bits + even parity
a b c d e f g P
Modulo 2 addition of:
all 'a' bits
all 'b' bits
all 'c' bits

When the data block transmitted was terminated by ETB the line is held so that further data of the same type can be sent when it becomes available. The blocks sent following an ETB must also contain the SOH and Header characters.

**Note:** ORION will accept as input concatenated data blocks, terminated by either ETB or ETX. However, it will only transmit discrete blocks, where the first two characters are SOH and Header and the termination character is ETX. This ensures that the line is only held by ORION for the duration of the transmission of a single block of data.

Flow charts in Fig. 10.1 and Fig. 10.2 show the BS 4505 protocol as adopted by ORION.

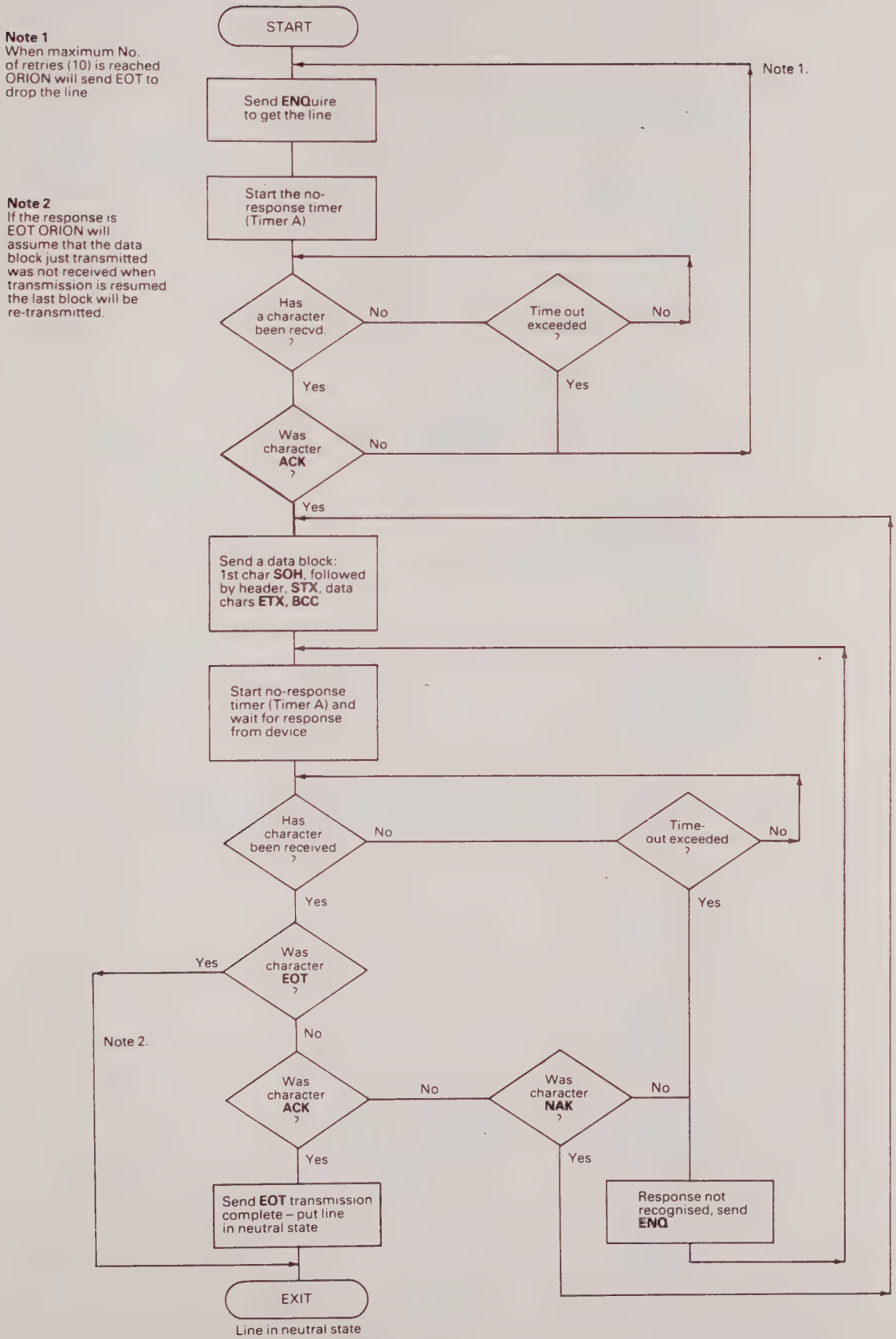

Note 1
When maximum No. of retries (10) is reached ORION will send EOT to drop the line
Note 2
If the response is EOT ORION will assume that the data block just transmitted was not received when transmission is resumed the last block will be re-transmitted.
START
Note 1.
Send ENQuire to get the line
Start the no-response timer (Timer A)
Has a character been recvd. ?
No
Time out exceeded ?
No
Yes
Yes
Was character ACK ?
No
Yes
Send a data block: 1st char SOH, followed by header, STX, data chars ETX, BCC
Start no-response timer (Timer A) and wait for response from device
Has character been received ?
No
Time-out exceeded ?
No
Yes
Yes
Was character EOT ?
Yes
Note 2.
No
Was character ACK ?
No
Was character NAK ?
No
Yes
Yes
Send EOT transmission complete – put line in neutral state
Response not recognised, send ENQ
EXIT
Line in neutral state

**Fig. 10.1** *ORION Transmit Protocol [BS4505]*

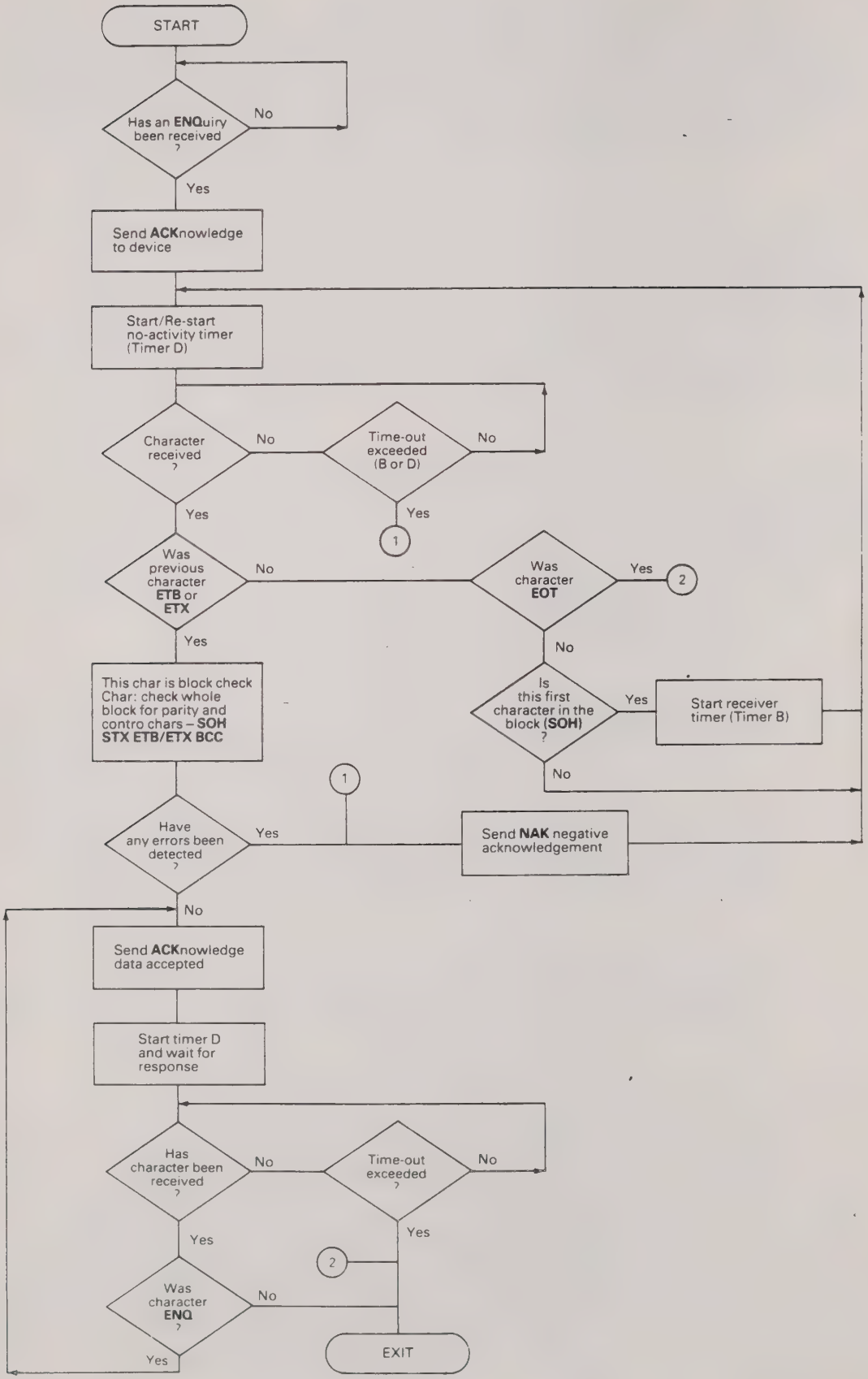
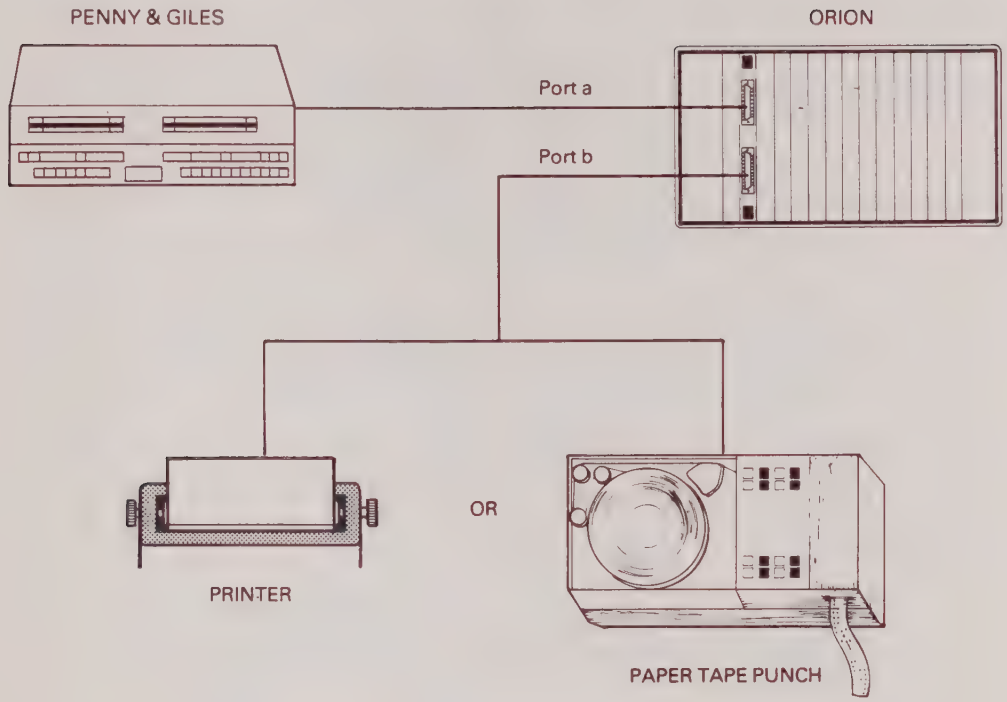



**Fig. 10.2** *ORION Receive Protocol [BS4505]*

## 3. OTHER RS232 PROTOCOL AVAILABLE




Ports A and B can, by switch selection, be used to drive output only devices. The output line termination/protocol available at each port is given below.

**Port A:** Configured to support Penny and Giles recorder protocol or a line termination of C/R.

**Port B:** Two line terminations available by switch selection:
C/R only, suitable interfacing to a cassette recorder or paper tape punch.
C/R LF 4 nulls, suitable for interfacing to a printer.

# 4. SERIAL INTERFACE CONNECTION DETAILS

**Pin details**

| PL1 and PL2 PIN NUMBERS | RS232 SIGNALS | RS422 SIGNALS |
|---|---|---|
| 2 | Tx | |
| 3 | Rx | Rx Hi (balanced) |
| 4 | RTS (REQUEST TO SEND) | |
| 7 | 0V | Rx Lo (balanced) |
| 11 | | Tx Lo (balanced) |
| 18 | | Tx Hi (balanced) |
| 20 | DTR (DATA TERMINAL READY) | |

**Baud Rate Settings**

| SW1 | | | |
|---|---|---|---|
| SWITCH CONTACTS | FUNCTION | SWITCH POSITION OFF | ON |
| 1 | SELECT BAUD RATE FOR PORT a (Continued on SW2) | SET ALL CONTACTS TO 'OFF' EXCEPT THE ONE BAUD RATE SELECTED | 153.6kB |
| 2 | | | 76.8kB |
| 3 | | | 38.4kB |
| 4 | | | 19.2kB |
| 5 | | | 9.6kB |
| 6 | | | 4.8kB |
| 7 | | | 2.4kB |
| 8 | | | 1.2kB |
| **SW2** | | | |
| 1 | SELECT BAUD RATE FOR PORT a (Continued from SW1) | SET ALL CONTACTS TO 'OFF' EXCEPT THE ONE BAUD RATE SELECTED | 600B |
| 2 | | | 300B |
| 3 | SELECT BAUD RATE FOR PORT b (Continued on SW3) | SET ALL CONTACTS TO 'OFF' EXCEPT THE ONE SELECTED | 153.6kB |
| 4 | | | 76.8kB |
| 5 | | | 38.4kB |
| 6 | | | 19.2kB |
| 7 | | | 9.6kB |
| 8 | | | 4.8kB |
| **SW3** | | | |
| 1 | SELECT BAUD RATE FOR PORT b (Continued from SW2) | SET ALL CONTACTS TO 'OFF' EXCEPT THE ONE BAUD RATE SELECTED | 2.4kB |
| 2 | | | 1.2kB |
| 3 | | | 600B |
| 4 | | | 300B |

**Protocol Selection**

| SW3 | | | |
|---|---|---|---|
| 5 | SELECT PARITY FOR PORT a (SW3/6 overrides SW3/5) | EVEN | ODD |
| 6 | | NO PARITY | PARITY |
| 7 | SELECT PARITY FOR PORT b (SW3/8) overrides SW3/7) | EVEN | ODD |
| 8 | | NO PARITY | PARITY |
| **SW4** | | | |
| 1 | SELECT RS232C/422 FOR PORT a | RS422 | RS232C |
| 2 | SELECT RS232C/422 FOR PORT b | RS422 | RS232C |
| 3 | SELECT PORT a PROTOCOL | (SW4/4 enabled) | BS4505 (overrides SW4/4) |
| 4 | | C/R | Penny & Giles protocol |
| 5 | SELECT PORT b PROTOCOL | (SW4/6 enabled) | BS4505 (overrides SW4/6) |
| 6 | | C/R | C/R, L/F and 4 NULLS |

## 4.1 CABLE FOR RS422 OPERATION

Cables listed in BICC Cableselector No E18 are suitable for use as a connection between equipment in normal indoor atmospheres. For use in more onerous conditions an armoured cable may be required.

The interconnecting cable characteristics as layed down in EIA standard for RS422 are given below:

**Conductor Size** The interconnecting cable shall be composed of two wires of a 24 AWG or larger conductor for solid or stranded copper wires, or for non-copper conductors, a sufficient size to yield a dc wire resistance not to exceed 30Ω per 1000 feet per conductor.

**Mutual Pair Capacitance** The capacitance between one wire in the pair to the other wire shall not exceed 20 pF per foot, and the value shall be reasonably uniform over the length of the cable.

**Stray Capacitance** The capacitance between one wire in the cable to all others in the cable sheath, with all others connected to ground, shall not exceed 40 pF per foot and shall be reasonably uniform for a given conductor over the length of the cable.

**Pair-to-Pair Balanced Crosstalk** The balanced crosstalk from one pair of wires to any other pair in the same cable sheath shall have a minimum value of 40 dB of attenuation measured at 150 kHz.

# Chapter 11
# GPIB Interface (IEEE 488)


| *Para. No.* | *Title* | *Page No.* |
|---|---|---|
| 1 | Introduction | 11.2 |
| 2 | GPIB status | 11.2 |
| 2.1 | Talker only status | 11.2 |
| 2.2 | Talker/Listener status | 11.2 |
| 3 | GPIB protocol | 11.3 |
| 4 | Service Request | 11.4 |
| 5 | Polling | 11.5 |
| 5.1 | Parallel Poll | 11.5 |
| 5.2 | Serial Pool | 11.5 |
| 6 | Input/Output data | 11.5 |
| 6.1 | Input data | 11.5 |
| 6.2 | Output data | 11.5 |
| 7 | The GPIB address | 11.5 |
| 8 | Communication via the GPIB bus | 11.6 |
| 9 | GPIB interface connection details | 11.7 |

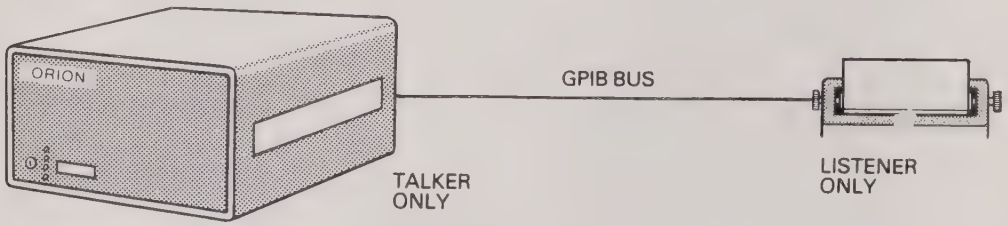
# 1. INTRODUCTION

The GPIB board, 35304A, fully implements the GPIB IEEE 488(1978) interface standard, and will support any GPIB compatible device such as a process controller, calculator or output peripheral. The GPIB standard permits up to 15 instruments to be connected to any one bus, of total bus length not exceeding 20 metres. Each instrument on the bus is designated as either a CONTROLLER or TALKER/LISTENER. ORION has the capability of being either a TALKER/LISTENER or a TALKER only device. The TALKER/LISTENER Status and address are switch selectable.

Both parallel polling and serial polling capabilities have been implemented and are described in section 5.

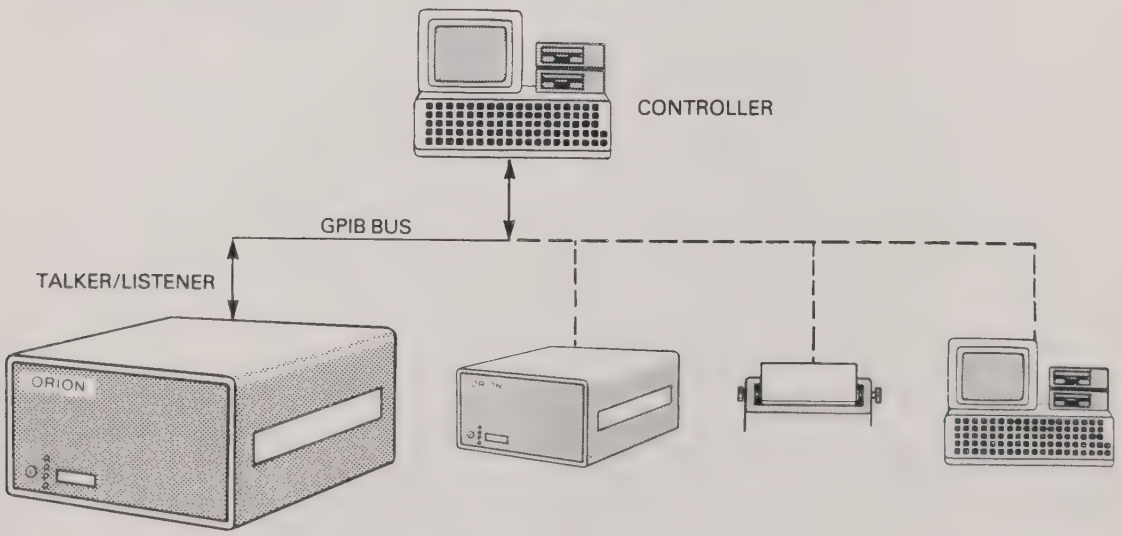
# 2. GPIB STATUS

## 2.1 TALKER ONLY STATUS




When ORION is designated TALKER only status it can be used to output data directly to a controller, as it becomes available, or alternatively to directly communicate with a high speed data recording device such as a printer or tape recorder. In the latter case, only the two devices occupy the bus, the data recording device is designated as a LISTENER only device and the ORION will communicate using data lines and Validation lines for hardware handshake.

## 2.2 TALKER/LISTENER STATUS




When ORION is attached to a GPIB bus for the purpose of control by a computer it must be designated TALKER/LISTENER status. As a TALKER/LISTENER ORION will be able to accept commands from another TALKER, eg a computer, and send back data to it, ie measurement data, error messages, command replies, etc.

## 3. GPIB PROTOCOL

The GPIB bus consists of 16 lines, which are subdivided as follows:

5 lines for bus MANAGEMENT

3 lines for VALIDATION of information transfer (handshake)

8 lines for DATA, these may carry information or instructions as defined by the MANAGEMENT lines.

**MANAGEMENT lines**

| Line | Use | Description |
|---|---|---|
| **IFC** | Interface Clear | Used by controller to put bus in known state |
| **REN** | Remote Enable | Used to indicate whether active listner will use programming information sent by user. |
| **ATN** | Attention | Indicates whether data line carries control information or data. |
| **EOI** | End of Identify | Asserted by talker to designate last byte **or** used in parallel poll. |
| **SRQ** | Service Request | Used by device to get attention of controller. |

**VALIDATION lines**

| Line | Use | Description |
|---|---|---|
| **DAV** | Data Valid | Asserted by talker when Valid data put on data line. |
| **NRFD** | Not Ready for Data | Held to ground by listener if not ready to receive data. |
| **NDAC** | No Data Accepted | Released by active listener when data has been accepted. |

**DATA lines**

DI01 to DI08 may carry either data or service request byte (SRQ) information, where DI01 is the least significant bit (bit 0).

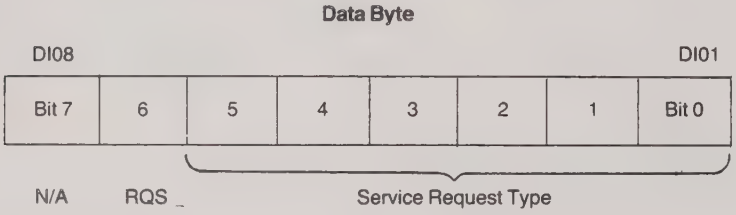
## 4. SERVICE REQUEST

When ORION wishes to send data to another device on the bus it must first gain the attention of the controller by sending a service request. This it does by setting the SRQ management line. In addition ORION will set up status information for the data lines DI01 to DI08 as follows:

| | |
|---|---|
| DI07 | set Request for service bit (RQS). |
| DI01 to DI06 | Service Request type data. |

In response to the request for service CONTROLLER may either serial poll or parallel poll ORION (and the other devices on the bus). A parallel poll would generally be used where there is more than one device on the bus that may require service in order to establish which device has issued the request. A serial poll may be used, by single or multi device systems, to interrogate the data byte for staus information on the requesting device, eg ORION.

The table below lists the ORION service request types together with the value of the resultant data byte. Bits 0 to 5 of the byte contain the service type, bit 6 is always set (RQS bit) and bit 7 is not used, so is always zero.




**Service Request Type**

| Reason for request | Value of data byte | | |
|---|---|---|---|
| | Binary | Decimal | Hex |
| Monitor Data available | 01000000 | 64 | 40 |
| Logged and/or processed data available. | 01000010 | 66 | 42 |
| Data copied from ORION cartridge available. | 01000100 | 68 | 44 |
| Message available, eg Power Fail/back, 'WRITE' data, run time error message, etc. | 01000110 | 70 | 46 |
| Listed data available (LIST command). | 01001000 | 72 | 48 |
| Command response data available (Query data, Error response, Immediate measurement results). | 01001010 | 74 | 4A |
| | Note | | |

**Note:** All other values of service request are invalid.

# 5. POLLING

## 5.1 PARALLEL POLL

The PPI sub set is followed in parallel polling. Before ORION will respond to a parallel poll command the response bit configuration for the ORION must have been set up remotely by the controller. This must be done using the parallel poll configuration command applicable to the controller being used.

When parallel polled by the CONTROLLER ORION will, if it requires service, respond by setting its designated response bit in the data byte. The CONTROLLER may then, if it wishes, establish the link for ORION to send its data. However, before establishing the link the controller may serial poll ORION to see what type of service request ORION has made.

## 5.2 SERIAL POLL

When the CONTROLLER serially polls ORION in response to a service request or parallel poll, it interrogates the status information set up by ORION in the data byte, DI01 to DI08. If ORION has sent a Service Request (SRQ) the Request for Service bit, bit 6, will have been set, and bits 0 to 5 will contain the type of service request. The CONTROLLER may now establish the link for ORION to send data.

# 6. INPUT/OUTPUT DATA

## 6.1 INPUT DATA

Input data command strings sent to ORION may be up to 80 character in length including terminating characters. The terminating characters may be either C/R LF or C/R only with EOI asserted on the final character.

**Note:** If the data string is 80 characters in length EOI need not be asserted.
If the data string is less than 80 characters in length EOI **must** be asserted on the final character.

## 6.2 OUTPUT DATA

Output from the ORION is in 20, 40 or 80 character blocks, terminated by linefeed, LF, with EOI asserted. The LF is additional to the 20, 40 or 80 characters in the block.

# 7. THE GPIB ADDRESS

The GPIB adddress is the means by which the CONTROLLER identifies each device on the bus. Each instrument on the bus is designated a unique address in the range zero to thirty. The ORION GPIB address is selected by the switches on the GPIB board, SW1/1 to SW1/5.

When selecting an address for an ORION care should be taken to ensure that the address is not the same as that of any other device on the bus.

| Error Message | Error Indicated |
|---|---|
| E1 Ovrld | Measurement overload |
| E2 Ovrld | Measurement out of linearisation range |
| E3 Ambnt | Invalid ambient temperature |
| E4 Ovrfl | Event total exceeds 65536 |
| E5 TmOut | Timeout in external module measurement (i.e. > 20 s) |
| E6 Initl | Strain gauge channel not initialised |
| E7 NoCom | Input connector not fitted |
| E8 Wait | Result not yet accumulated (Counter Timer Module) |
| E9 Off | Counter Timer Channel is off |

Error messages are included in Full Format and Compact Format results, see the examples below.

e.g. 1. Typical error

## 8. COMMUNICATION VIA THE GPIB BUS

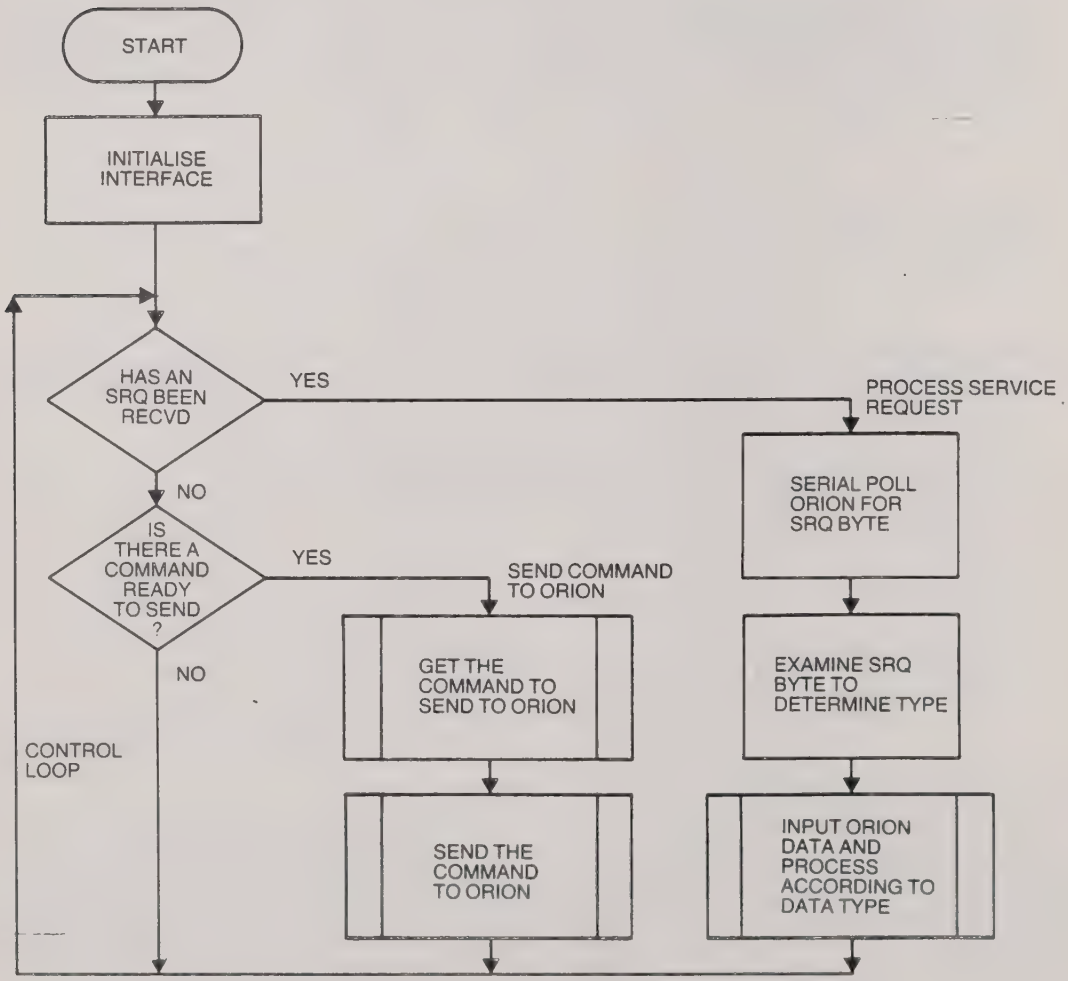


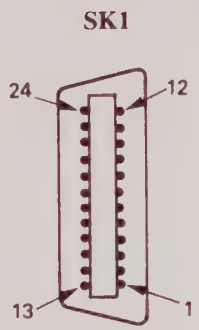

**Fig 11** *GPIB Communications Flow Chart.*

Figure 11 is an example of a simple flow chart for a routine that could be used to read data from and send commands to an ORION, via the GPIB bus.

# 9. GPIB INTERFACE CONNECTION DETAILS

SK1 Pin Details

| Pin No | Signal |
|---|---|
| 1 | DIO1 |
| 2 | DIO2 |
| 3 | DIO3 |
| 4 | DIO4 |
| 5 | EOI |
| 6 | DAV |
| 7 | NRFD |
| 8 | NDAC |
| 9 | IFC |
| 10 | SRQ |
| 11 | ATN |
| 12 | |
| 13 | DIO5 |
| 14 | DIO6 |
| 15 | DIO7 |
| 16 | DIO8 |
| 17 | REN |
| 18 | 0V |
| 19 | 0V |
| 20 | 0V |
| 21 | 0V |
| 22 | 0V |
| 23 | 0V |
| 24 | 0V |




SW1 Switch Settings

| Switch Contacts | Function | Switch Position Off | Switch Position On |
|---|---|---|---|
| 1<br>2<br>3<br>4<br>5 | Talker/Listener Address | Address 0 to 31, See Table Below | |
| 6 | Talker/Listener Status | Talker only | Talker/ Listener |
| 7 | Not used | – | – |
| 8 | Not used | – | – |

| | GPIB ADDRESS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| SW1/1 | ON | OFF | ON | OFF | ON | OFF | ON | OFF | ON | OFF | ON | OFF | ON | OFF | ON | OFF | ON | OFF | ON | OFF | ON | OFF | ON | OFF | ON | OFF | ON | OFF | ON | OFF | ON | OFF |
| SW1/2 | ON | ON | OFF | OFF | ON | ON | OFF | OFF | ON | ON | OFF | OFF | ON | ON | OFF | OFF | ON | ON | OFF | OFF | ON | ON | OFF | OFF | ON | ON | OFF | OFF | ON | ON | OFF | OFF |
| SW1/3 | ON | ON | ON | ON | OFF | OFF | OFF | OFF | ON | ON | ON | ON | OFF | OFF | OFF | OFF | ON | ON | ON | ON | OFF | OFF | OFF | OFF | ON | ON | ON | ON | OFF | OFF | OFF | OFF |
| SW1/4 | ON | ON | ON | ON | ON | ON | ON | ON | OFF | OFF | OFF | OFF | OFF | OFF | OFF | OFF | ON | ON | ON | ON | ON | ON | ON | ON | OFF | OFF | OFF | OFF | OFF | OFF | OFF | OFF |
| SW1/5 | ON | ON | ON | ON | ON | ON | ON | ON | ON | ON | ON | ON | ON | ON | ON | ON | OFF | OFF | OFF | OFF | OFF | OFF | OFF | OFF | OFF | OFF | OFF | OFF | OFF | OFF | OFF | OFF |

# Appendix 1
# Error Messages

ORION will give automatic indication of detected errors. Programming errors and other faults are indicated by error messages on the Display, and measurement errors are included in the output results.

## 1 PROGRAMMING ERRORS AND OTHER FAULTS INDICATED ON THE DISPLAY

| Error No. | Error | Remedy |
|---|---|---|
| 001 | Unrecognised command | Re-enter, starting at the beginning of the function. |
| 003 | No query available | The requested information is not available. |
| 021 | Illegal numeric input | Re-enter numbers correctly. |
| 031 | Illegal command | Re-enter starting at beginning of the function. |
| 041 | Illegal option | Re-enter starting at beginning of the function. |
| 061 | Strain gauge initialisation not possible | Check for damaged, or over-strained gauge. Check for broken connection. |
| 091 | Cartridge out/Tape fault | Replace cartridge, or check for broken or damaged tape. (3530D only) |
| 092 | Cartridge write-protected | Select 'Record' on cartridge, or, to avoid overwriting on this cartridge, fit another set to 'Record'. (3530 only) |
| 093 | Cartridge data error: tape in illegal position | Tape is possibly faulty, fit new cartridge. (3530D only) |
| 094 | Cartridge data error: tape past end of tape holes | |
| 095 | Cartridge data error: Software time out | |
| 096 | Cartridge data error: short block detected | Recorded data may have been corrupted by magnetic field; data should be re-recorded. or Tape is possibly faulty, fit new cartridge. (3030D only) |
| 097 | Cartridge data error: CRC error detected | |
| 098 | Cartridge data error: Incorrect format on tape | |
| 099 | Tape system busy (eg copying is in progress) | Wait until tape system is free. (3530D only) |

| Error No. | Error | Remedy |
|---|---|---|
| 110 | Not in local control | Press LOCAL and wait for local control to be granted, or use remote device to relinquish control of ORION. |
| 113 | Local control not available | Disable local lock-out condition on remote controller. |
| 118 | Logger running | Program cannot be changed while logger is set to RUN, select HAlt if necessary. |
| 119 | Monitor active | Program cannot be changed while logger is Monitoring; de-select Monitor if necessary. |
| 150 | Processing: illegal line number | Enter correct line number. |
| 151 | Processing: no room for program | Reduce program length. |
| 152 | Processing: no room for variable | Reduce program length. |
| 153 | Processing: illegal element in a line of program | Re-read the relevant section in Chapter 6 to ascertain correct form |
| 154 | Processing: FOR/NEXT nesting error | |
| 155 | Processing: IF/THEN/ELSE/ ENDIF nesting error | |
| 156 | Processing: Syntax error | |
| 157 | Processing: line too long | Reduce line length |
| 158 | Processing: parenthesis depth error | |
| 159 | Processing: illegal variable name | Re-read the relevant section in Chapter 6 to ascertain correct form. |
| 160 | Processing: variable name too long | |
| 161 | Processing: resequence failed, ie a request to insert extra line numbers has been rejected | |
| 162 | Processing: line does not exist | Check line numbers. Error may be part of a GOTO instruction |
| 171<br>172 | The appearance of these codes in the Display indicates a fault condition | The logger requires servicing. |
| 173 | Run Time: index error eg (I) = 0, or-ve or >600 | Replace with a permissible index number. |

| Error No. | Error | Remedy |
|---|---|---|
| 174 | Run Time: channel not present | Either correct the channel number in the program, or add the channel to the Task Definition. |
| 175 | Run Time: formula too complex, ie more than 4 pending operations | Split the formula into two two separate lines of program. |
| 199 | As for 171 and 172 | Logger requires servicing. |
| 253 | No energisation module | A Transducer Energisation Module (35301D) must be fitted in order to measure the sensor type specified. |
| 254 | Channel number conflict | The channel numbers specified do not correspond (wholly or partially) with the type or quantity of Input Selectors fitted. Enter the correct channel numbers. |
| 255 | Function not applicable (eg Alarms not applicable to event channels) | The requested function is not available. |

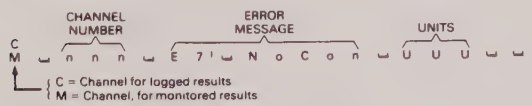
## 2 MEASUREMENT ERRORS INCLUDED IN THE OUTPUT RESULTS

If an error is detected, one of the following 9 error messages will be output instead of the numerical value of the result.

| Error Message | Error Indicated |
|---|---|
| E1 Ovrld | Measurement overload |
| E2 Ovrld | Measurement out of linearisation range |
| E3 Ambnt | Invalid ambient temperature |
| E4 Ovrfl | Event total exceeds 65536 |
| E5 TmOut | Timeout in external module measurement (ie > 20 s) |
| E6 Initl | Strain gauge channel not initialised |
| E7 NoCom | Input connector not fitted |
| E8 Wait | Result not yet accumulated (Counter Timer Module) |
| E9 Off | Counter Timer Channel is off |

Error messages are included in Full Format and Compact Format results, see the examples below.

eg 1. Typical error message in a Full Format result.




eg 2. Typical error message in a Compact Format result.

The 9 error messages E1-E9 correspond with the 9 error codes in the Tag Bytes of Coded Format results, see page 7.8.

## 3 TAPE HANDLING MESSAGE

When replaying data from the ORION cartridge one of the following messages may be output to the controlling device.

| Messages | Reason |
|---|---|
| *old format | Tape not recorded on 3530D ORION |
| *tape fault | Cartridge not in reader |
| *write protected | Cartridge write protected |
| *end of tape | End of tape reached |
| *end of data | End of data reached |
| *short block | Recorded data corrupted or faulty tape |
| *CRC error | |
| *bad format | |

**Note:** The first character of the error message will be a space character, the second character will be an asterisk.

# Appendix 2
# Sensor Number Tables

**SENSOR NUMBERS FOR EVENT STATUS AND DIGITAL I/O**

| 1st DIGIT | | 2nd DIGIT | | 3rd DIGIT | | NOTES |
|---|---|---|---|---|---|---|
| 7 | Event status card | 0 | BIT pattern input | 1 to 8 | Number of bits grouped | For individual or grouped status inputs, allows a number of digital levels to be examined by a scanning task for e.g. alarm condition. |
| | | 1 | BCD | 1 to 5 | Number of decades | For BCD inputs, each BCD decade uses 4 consecutive input channels, these are converted by ORION into a decimal number. |
| | | 2 | Inverted BCD | 1 to 5 | Number of decades | For inverted (negative logic) BCD inputs. |
| | | 4 | Totalise | 0<br>1<br>2 | Hi-Lo<br>Lo-Hi<br>Both | Cumulative total of input pulses (transitions)<br>Maximum total = 65535 |
| | | 5 | Count | 0<br>1<br>2 | Hi-Lo<br>Lo-Hi<br>Both | Count of input pulses for scan. Count reset at start of each scan. |
| | | 6 | Event capture | 0<br>1<br>2 | Hi-Lo<br>Lo-Hi<br>Both | Channels continuously monitored for 'events'. All events detected are regarded as alarms. |
| | | 7-9 | Not used | | | |
| 8 | Digital I/O card | 0 | BIT pattern | 1 to 8 | Number of bits | For individual or grouped status inputs. Inputs and outputs are used independently. Inputs provide status information, outputs give alarm or control (stimulus) information. |
| | | 1 | BCD | 1 to 6 | Number of decades | External module measurement (port mode): can be used to read binary or BCD data from external modules. |
| | | 2 | Inverted BCD | 1 to 6 | Number of decades | |
| | | 3 | Binary | 1 to 3 | Number of 8 bit bytes | |

## SENSOR NUMBERS FOR REED OR SOLID STATE SELECTOR

| 1st DIGIT | | 2nd DIGIT | | 3rd DIGIT | | | NOTES |
|---|---|---|---|---|---|---|---|
| 1 | Volts | 1 | DC | 0 | Auto range | Range | |
| | | | | 1 | 10mV | | |
| | | | | 2 | 100mV | | |
| | | | | 3 | 1V | | |
| | | | | 4 | 10V | | |
| | | | | 5 | 100V | | |
| | | 2 | AC | 0 | Auto range | Range | |
| | | | | 1 | 20mV | | |
| | | | | 2 | 200mV | | |
| | | | | 3 | 2V | | |
| | | | | 4 | 20V | | |
| | | | | 5 | 20V | | |
| 2 | Resistance | | 4-terminal | 0 | Auto range | DC voltage Range without thermal drift compensation | Result will be in Ω, except when current is user defined. In this case a 100μA current is assumed and conversion must be applied to the result. |
| | | 1 | 1mA | 1 | 10mV | | |
| | | 2 | 100μA | 2 | 100mV | | |
| | | 3 | User defined | 3 | 1V | | |
| | | | 2-terminal | 4 | 10V | | |
| | | 5 | 1mA | 5 | Auto range | Range with thermal drift compensation | |
| | | 6 | 100μA | 6 | 10mV | | |
| | | 7 | User defined | 7 | 100mV | | |
| | | | | 8 | 1V | | |
| | | | | 9 | 10V | | |
| 3 | Thermocouple | 1 | Type E | 0 | Ambient | Thermocouple reference junction compensation | |
| | | 2 | Type J | 1 | Ext Ref 1 | | |
| | | 3 | Type K | 2 | Ext Ref 2 | | |
| | | 4 | Type R | 3 | Ext Ref 3 | | |
| | | 5 | Type S | 4 | Ext Ref 4 | | |
| | | 6 | Type T | 5 | Ext 0°C | | |
| 4 | Platinum resistance thermometer | | 4-terminal | 0 | Auto range | Range without thermal drift compensation | |
| | | 1 | 1mA | 1 | 10mV | | |
| | | 2 | 100μA | 2 | 100mV | | |
| | | 3 | User defined | 3 | 1V | | |
| | | | 2-terminal | 4 | 10V | | |
| | | 5 | 1mA | 5 | Auto range | Range with thermal drift compensation | |
| | | 6 | 100μA | 6 | 10mV | | |
| | | 7 | User defined | 7 | 100mV | | |
| | | | | 8 | 1V | | |
| | | | | 9 | 10V | | |
| 5 | Current | 1 | DC | 0 | Auto range | Range assuming 100Ω shunt | |
| | | | | 1 | 100μA | | |
| | | | | 2 | 1mA | | |
| | | | | 3 | 10mA | | |
| | | | | 4 | 100mA | | |
| | | 2 | AC | 0 | Auto | Range assuming 100Ω shunt | |
| | | | | 1 | 200μA | | |
| | | | | 2 | 2mA | | |
| | | | | 3 | 20mA | | |
| | | | | 4 | 200mA | | |
| | | | | 5 | 200mA | | |
| 6 | Strain gauge | | ½ or ¼ bridge (dual current) | 0 | ½ bridge | Without thermal drift compensation | |
| | | | | 1 | ¼ bridge 120Ω | | |
| | | | | 2 | ¼ bridge x | | |
| | | | | 3 | ¼ bridge y | | |
| | | | | 4 | ¼ bridge z | | |
| | | 0 | 8.33mA | 5 | ½ bridge | With Thermal drift compensation | |
| | | 1 | 1mA | 6 | ¼ bridge 120Ω | | |
| | | 2 | 100μA | 7 | ¼ bridge x | | |
| | | 3 | User defined | 8 | ¼ bridge y | | |
| | | | | 9 | ¼ bridge z | | |
| | | | full bridge | | | | |
| | | 4 | 8.33mA | 0 | full bridge | No thermal drift comp. | |
| | | 5 | 1mA | | | | |
| | | 6 | 100μA | | | | |
| | | 7 | User defined | | | | |
| | | 8 | 2.0 volts | 5 | full bridge | Thermal drift compensation | |

## SENSOR NUMBERS FOR COUNTER TIMER MODULE

| 1st DIGIT | | 2nd DIGIT | | 3rd DIGIT | | NOTES |
|---|---|---|---|---|---|---|
| 9 | Counter timer module | 0 | Frequency | 0 | External gate | |
| | | | | 1 | 1 sec (Internal gate) | |
| | | | | 2 | 1 sec (Internal gate) | |
| | | | | 3 | 10 sec (Internal gate) | |
| | | | | 4 | 100 sec (Internal gate) | |
| | | 1 | Totalise | 0 | External Gate | |
| | | 2 | Period | 1 | 1 (Number of periods averaged) | |
| | | | | 2 | 10 (Number of periods averaged) | |
| | | | | 3 | 100 (Number of periods averaged) | |
| | | | | 4 | 1000 (Number of periods averaged) | |
| | | | | 5 | 10,000 (Number of periods averaged) | |
| | | 3 | Interval | 0 | Internal gate | |
| | | 4 | off | 0 | Counter timer function off | |
| 0 | Counter timer module | 4 | off | 0 | Counter timer function off | May be used instead of 940 to set counter timer function off |

## SENSOR NUMBER FOR ANALOGUE OUTPUT MODULE

An analogue output channel must have the sensor number 999 in order that it may be 'scanned' or 'monitored'.

# Appendix 3
# Example Program – ORION/HP85 RS232 Interface

The following example program can be used to control ORION when connected to an HP85 calculator via the RS232 Interface on the ORION controller board. Using the HP-85 function keys the user is able to put the ORION into COMMAND mode and send setup, measurement, programming commands etc. to ORION. Whilst in RESULT mode the program will display ORION data on the HP85 screen, scrolling the data up the screen when it becomes full.

```
10    ! RS3530
20    DIM A$[1],D$[81],S$[21],K$[80]
30    !
40    GOSUB 920 ! Initialise
50    IF D=1 THEN GOSUB 140 ! Data available
60    IF X<>0 AND D=0 THEN GOSUB 390 ! Send command
70    IF K=1 THEN GOSUB 330 ! Function key pressed
80    GOTO 50
90    !
100   CLEAR
110   DISP USING "5/,K" ; "Program Terminated"
120   END
130   !
140   I=CEIL(LEN(D$)/32)*32
160   IF C+I<=416 THEN 200
170   CLEAR
180   C=0
190   KEY LABEL
200   IF X=2 THEN DISP USING 210; D$ @ GOTO 260
210   IMAGE //,"ORION Error :- ",K
220   IF X=3 THEN DISP USING 230 ; D$ @ GOTO 260
230   IMAGE /"ORION Response :- ", /,K
240   DISP D$
250   C=C+I
260   D=0
270   IF X=1 THEN X=0 @ GOTO 310 ! Display Data
280   ON KEY# 1, "Continue" GOTO 310
290   KEY LABEL
300   GOTO 300
310   ENABLE INTR S;M @ RETURN
320   !
330   OFF KEY# 1
340   L=1 ! Command Mode
350   K=0
360   OUTPUT S USING "#,A" ; "!"
370   RETURN
380   !
390   CLEAR
400   C=0
```

```
410   OFF KEY# 1
420   DISP USING 430 ; S$
430   IMAGE 4X,21A,2/,2X,"To return to 'RESULTS mode'",5X,"enter a null
      command",//,"Enter command"
440   INPUT K$
450   K$=UPC$(K$)
460   IF K$<>"" THEN OUTPUT S USING "K" ; K$ @ GOTO 530
470   ! Release Line
480   CLEAR
490   L=0
500   ON KEY# 1, "Command" GOSUB 560
510   KEY LABEL
520   OUTPUT S USING "#,A" ; "#"
530   X=0
540   RETURN
550   !
560   BEEP @ K=1 @ RETURN
570   !
580   ENTER S USING "#,K" ; A$
590   X=POS("OERNLCP",A$)
600   IF X=0 THEN DISP "INVALID PROMPT ";A$ @ GOTO 860
610   ON X GOTO 630,630,630,680,720,760,800
620   ! O,E & R Prompt
630   OUTPUT S USING "#,A" ; "?"
640   ENTER S USING "%,%K" ; D$
650   D=1 ! Set data available flag
660   GOTO 860
670   ! N Prompt
680   IF L=1 THEN OUTPUT S USING "K" ; "%1" @ GOTO 850 ! Send "%1"
690   S$="Device not recognised"
700   GOTO 850
710   ! L Prompt
720   IF L=1 THEN OUTPUT S USING "K" ; "%4" @ GOTO 850 ! Send "%4"
730   S$="LOCAL control"
740   GOTO 850
750   ! C Prompt
760   IF L=1 THEN L=2
770   S$="FULL control"
780   GOTO 850
790   ! P Prompt
800   CLEAR
810   DISP USING "5/,K" ; "Communication Prohibited"
820   ON KEY# 1, "Continue" GOTO 850 ! Return to results mode
830   KEY LABEL
840   GOTO 840
850   ENABLE INTR S;M
860   RETURN
870   !
880   CLEAR
890   DISP USING "5/,K"; " Serial interface has exceeded timeout period"
900   END
910   !
920   CLEAR
930   !
940   D=0
950   I=0
960   K=0
```

```
970   L=0
980   S=0
990   C=0
1000  X=0
1010  M=16
1020  T=5000 ! Period allowed before timeout (5 seconds)
1030  !
1040  ON ERROR GOTO 1060
1050  A=BTD("1") @ GOTO 1080
1060  OFF ERROR @ CLEAR @ DISP USING "5/,K" ; "No I/O ROM" @ BEEP @ STOP
1070  !
1080  ON ERROR GOTO 1120
1090  FOR I=3 TO 10
1100  STATUS I,0 ; A
1110  IF A=2 THEN S=I @ I=10
1120  NEXT I
1130  IF S=0 THEN BEEP @ CLEAR @ DISP "No RS-232 interface 1 in system" @
      STOP
1140  OFF ERROR
1150  RESET S
1160  !
1170  CONTROL S,2 ; BTD("00000111")
1180  !
1190  CONTROL S,3 ; 15 ! 9600 Baud
1200  !
1210  CONTROL S,4 ; BTD("00011010")
1220  !
1230  CONTROL S,5 ; BTD("00000000")
1240  !
1250  CONTROL S,9 ; BTD("10100101")
1260  !
1270  CONTROL S,11 ; BTD("00000010")
1280  !
1290  CONTROL S,12 ; 13 ! Carriage return
1300  !
1310  CONTROL S, 16 ; 1 ! 1 character
1320  !
1330  CONTROL S,17 ; 13 ! carriage return
1340  !
1350  SET TIMEOUT S;T
1360  ON TIMEOUT S GOTO 880
1370  ON KEY# 1, "Command" GOSUB 560
1380  ON KEY# 4, "Exit" GOTO 100

1390  KEY LABEL
1400  ON INTR S GOSUB 580
1410  OUTPUT S USING "#,A" ; "#"
1420  J$=";,"
1425  CONVERT OUT 10 PAIRS ; J$
1430  ENABLE INTR S;M @ RETURN
```

## 1 HARDWARE

The minimum hardware configuration on which the example program will run is as follows:

1. HP-85 fitted with:

| | |
|---|---|
| RS232 Serial Interface | 82939A |
| ROM drawer | 82936A |
| I/O ROM | 0085-15003 |

2. One 3530D or 3530F ORION fitted with standard RS232 controller card in slot 1.

## 2 OPERATING INSTRUCTIONS

Set up the ORION as follows:

On the controller board set
baud rate to 9.6kB SW1/1 to 7 off, SW1/8 on.
Computer/calculator mode SW2 S1 OFF, S2 and S3 ON.

Type in the program at the keyboard, or load from a previously saved HP85 tape.

When program is in memory press the RUN key on the HP85.

When the program is running the VDU screen will clear, the lower section of the screen will display the prompt:

```
. . . . . .
command                    exit
```

where: *command* is positioned over function key 1,
*exit* is positioned over function key 4.

To send command data to ORION press function key 1.
To terminate the program press function key 4.

**Note:** If at the time the program is run a task is already running on ORION, where the output had been directed to the RS232 port, S0, the data will be displayed in the area of screen above the function key prompts.

**Send Commands:** Press function key 1 to gain control of the ORION RS232 link. The display of results on the VDU, if in progress, will be suspended. The VDU screen will clear and the following prompt displayed:

*Full control*

*To return to 'RESULT' mode*
*enter a null command*

*Enter command*
*?*

You are now in full control of ORION, commands can be entered to set up, amend or interrogate the ORION channel and task database, or program the ORION.

To send a command type the command string followed by a carriage return (C/R), and the string will be sent to ORION.

**Note:** When entering command strings or program lines that contain commas (,) the comma **must** be typed as a semicolon (;). However when the string is sent to ORION it will be sent as a comma (,) by the program.

If the command string was in error the message

*ORION Error:- E nnn*

will be displayed function key 1 must be pressed before the command string may be re-entered.

If the command sent required ORION response eg Query or immediate measurement commands, the message

*ORION Response:-*

*aaaaaaa*

will be displayed. Function Key 1 must be pressed before the next command string may be entered.

To exit from the command mode, a null string must be entered, ie C/R only, and return will be made to 'RESULT' mode.

**Display results:** To display results data ORION must be in 'RESULT' mode. When in results mode results directed to the RS232 port (S0) will be displayed, in addition the function key prompt will be displayed at the bottom of the screen.

**Terminate program:** press function key 4 and the message:

*Program Terminated*

will be displayed. Pressing function key 4 will terminate the program in both 'COMMAND' and 'RESULT' mode.

**Error Messages**

A number of error messages may be displayed whilst the program is running. The table below lists the messages, possible cause, and remedial action required.

**ERROR MESSAGES**

| HP-85 Display | Cause | Action |
|---|---|---|
| Communication Prohibited | A higher priority device has control of ORION, eg a GPIB controller | Relinquish control of other device, press Function Key 1 to continue |
| No I/O ROM | I/O ROM not inserted in HP-85 slot | Insert I/O ROM in HP-85, rerun program. |
| No RS232 Interface in system | HP-85 RS232 Interface missing | Insert HP-85 RS232 Interface, rerun program |
| Serial Interface has exceeded time out period | Power fail on ORION or communication error | Rerun program |

## 3 EXAMPLE PROGRAM BREAKDOWN

The function of each of the routines used in the example program has been described in the pages that follow. A line by line breakdown has been given where applicable.

| Line Nos. | |
|---|---|
| 40 | Go to Initialisation routine at line 920 |
| 50 | If data available go to subroutine at 140 |
| 60 | If command to be sent go to subroutine at 390 |
| 70 | If Function Key 1 pressed, COMMAND, go to 330 |
| 80 | else loop waiting for action |
| | |
| **100-120** | **Termination routine** |
| 100 | Clear screen |
| 110 | Display 'Program terminated' |
| | |
| **140-310** | **Data available routine** |
| 140 | Calculate length of input string |
| 160-190 | If insufficient space on screen to display data clear the screen. |
| 200-210 | Error response display |
| 220-230 | ORION command response display |
| 260 | Clear data available flag |
| 270 | If data received whilst in RESULT mode don't pause, else |
| 300 | Wait for function key to be passed |
| 310 | Enable interrupts and return |
| | |
| **330-370** | **Function Key 1 routine** |
| 330 | Disable function key interrupts |
| 340 | Set line status to 1 to indicate that a request to enter COMMAND mode has been made |
| 350 | Acknowledge that an attempt is being made to enter command mode |
| 360 | Send a '!' to ORION to enter COMMAND mode |
| 370 | Return |
| | |
| **390-540** | **Send a command routine** |
| 390-430 | Clear screen and display prompt to enter command |
| 440-450 | Input command, convert lower case characters to upper case |
| 460 | If not a null string send it to ORION and go to line 530, else |
| 470-510 | Clear screen, change line status to RESULT mode, re-enable function key interrupts |
| 520 | Send "#" to ORION to return to RESULT mode. |
| 530-540 | Clear old prompt and return |
| | |
| **560** | **Function key interrupt, flag function key pressed** |
| | |
| **580-860** | **ORION Interrupt Service Routine** |
| | |
| 580-590 | Get the ORION prompt and check validity |
| 610 | If valid prompt go to appropriate routine |
| 630 | O, E or R prompt, data available, send ? to request data |
| 640-660 | Input a line of data and set data available flag |
| 670-700 | N prompt, send %1 to ORION to identify computer as controlling device |
| 710-740 | L prompt, send %4 to computer in full control |
| 790-780 | C Prompt, Computer in full control set flag |
| 790-840 | P Prompt, a higher priority device has control, display message and wait for function key 1 to be pressd to continue |
| 850-860 | Re-enable interrupts and return |
| 880-900 | Serial Interface timeout interrupt routine, display time out message and halt |

**920-1410** **Initialisation routine**

940-1020 Define variables
D – Data available flag
I – Temporary variable
K – Function – Key flags
L – Line status 0 = RESULT mode
S – Variable for serial interface address
C – Display line count
X – Prompt
M – 16 Interrupt enable mask, interrupt on received data
T – 5000 Period allowed before timeout (5 seconds)

**1040-1060** **Locate I/O ROM, if not present beep and stop**

1080-1150 Determine serial interface address
1100 Status reg.0 contains the interface identifier
1130 If not present beep and stop

**1170-1330** **Configure interface options**

1170 Active DSR, DCD, CTS
1190 Baud rate 9.6kB
1210 Even parity, 7 bits data, One stop bit
1230 Set modem features
1250 Enable transmitter, strip received null, reset receive queue, Enable receiver
1270 Terminate input if character matches that specified in register 12 (C/R)
1290 Terminate on carriage return (C/R)
1310 1 EOL character
1330 EOL character is C/R

1350 Define timeout period for serial interface
1360 Routine to be executed on timeout at line 880
1370 Goto subroutine at 560 on function key 1 (COMMAND)
1380 Go subroutine at 100 on function key 4 (terminate)
1390 Label function keys
1400 Direct ORION interrupts to routine at line 580
1410 Send a # character to put the interface in a known state
1430 Enables interrupts and return
1420-1425 Output char ';' as a ','

# Appendix 4
# Example Program
# – ORION/HP85 via GPIB

The following example program can be used to control ORION when connected to an HP85 calculator via the GPIB IEEE 488 interface. A standard command string is sent to ORION and 'RUN', the data is displayed on the HP85 screen. Using the function keys the user may halt the display of data to send further command strings to ORION.

```
05        ! GP3530
10        DIM B$[80]
20        CONTROL 7,16;130,13,10
30        ON INTR 7 GOSUB 200
40        REMOTE 7
50        CLEAR 701
60        WAIT 4000
65        CLEAR
70        ENABLE INTR 7;8
75        A$ = ";,"
80        CONVERT OUT 7 PAIRS ; A$
85        OUTPUT 701;"CH 1-20 SE 110"
90        OUTPUT 701;"TA 1 OP ME TR TI DE 5 CO* RE IN IN 5"
100       OUTPUT 701;"TA1 CH1-20 AT F LO EV FO CO MA EV TO GP"
110       OUTPUT 701;"RUN"
120       !LOOP WAITING FOR DATA
130       ON KEY# 1,"COMMAND" GOSUB 300
140       ON KEY# 4,"EXIT" GOSUB 390 @ KEY LABEL
150       GOTO 150
200       !SERVICE ROUTINE
210       STATUS 7,1;A
220       S=SPOLL (701)
230       !
240       !READ ORION INPUT
250       ENTER 701;B$
260       DISP B$
270       ENABLE INTR 7;8
280       RETURN
290       !
300       ! SEND ORION COMMAND ROUTINE
310       OFF INTR 7
320       CLEAR
330       DISP "ENTER ORION COMMAND"
340       INPUT B$
350       OUTPUT 701;B$
360       ON INTR 7 GOSUB 200
370       RETURN
380       !
390       CLEAR
400       DISP USING "5/,K" ; "Program terminated"
410       END
```

## 1 HARDWARE

The minimum hardware configuration on which the exampe program will run is as follows:

1. HP-85 fitted with: HP-IB Interface 82937A
   ROM Drawer 82936A
   I/O ROM 0085-15003

2. One 3530D or 3530F fitted with:
   Analogue input card in Slot 4
   GPIB Card in slot 2 (35304A)
   Controller card in slot 1

## 2 OPERATING INSTRUCTIONS

Set up the ORION as follows:

One 3530D or 3530F fitted with:

**Controller Board:** SW/2 OFF – Boot Mode disabled

**Note:** When automatic loading is required, the delay in program line 60 must be increased in order to allow time to load a tape program from cartridge.

**GPIB Interface:** SW1/1 OFF Select talker listener address of 1
SW1/2 to 5 ON

SW1/6 ON Select talker/listener status

Type in the program at the keyboard, or load from a previous saved HP-85 tape.

When program is in memory type

RUN

The program will send a preset command string to ORION, via the GPIB Interface, which will set up and run a measurement task. The result data will be displayed on the HP-85 screen. The lower section of the screen will display the prompt:

---

*COMMAND* *EXIT*

where: *COMMAND* is positioned over function Key 1
*EXIT* is positioned over function Key 4

By pressing function Key 1 the user may, whilst the program is running, halt the display of ORION data and send a command string, of his own choice, to the ORION. Function Key 4 will terminate the program.

The standard task sent at the start if the program is a voltage measurement task, measuring channel 1 to 20 and outputting the results on the HP-85 screen in compact ASCII format.

**Sending ORION Commands**

To stop the display of data and enter an ORION command string press function Key 1.

The HP-85 will respond by clearing the screen and displaying the prompt:

*ENTER ORION COMMAND*

Type in a command string terminated by a carriage return (C/R) and the command will be sent to ORION.

**Note:** When typing command strings or ORION program lines, a comma must be typed as a semicolon (;). However when the string is sent to ORION it will be sent as a comma by the program.

The ORION response will be dependant on the command sent and its validity.

For a valid command: ORION will action the command, the HP-85 will display the command response, where applicable. If the command sent was not to halt the task or redirect the output, the task data will once again be displayed on the HP-85 screen.

For a command in error: ORION will ignore the command an ' the ORION error number will be displayed on the HP-85 screen. The display of task data on the HP-85 screen will be resumed if appropriate.

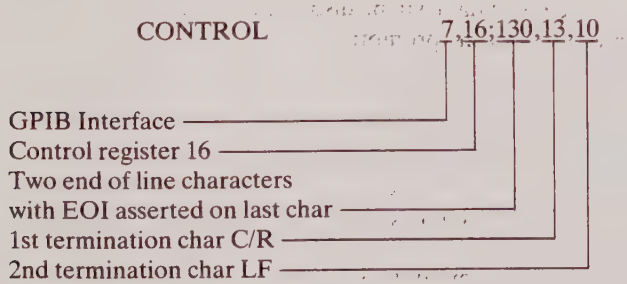
Whenever a new command string is to be sent to ORION function key 1 must be pressed at the HP-85, before the command string may be input.

## 3 EXAMPLE PROGRAM BREAKDOWN

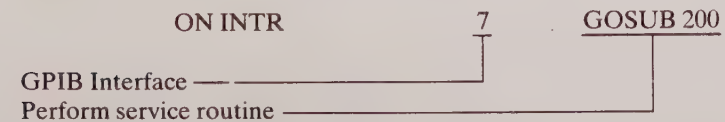
The BASIC statements that have been used in the example program are described below.

**Program lines 10 to 80:** perform the initialisation of the interface.

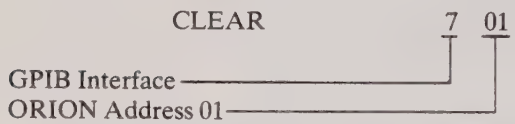
**line 20:** establishes the termination control characters C/R LF with EOI asserted.




Line 30 establishes the action to be taken on receiving an interrupt, ie perform subroutine at line 200.

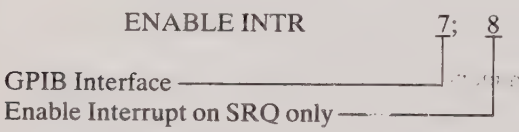


Line 50 sends a device clear to the ORION which will cause the ORION to give a Reset and Power back message.




Line 60 provides a delay of approximately 4 seconds, before continuing with the program. This delay is sufficient if no tape program is to be automatically loaded, ie auto boot mode off. However if Auto boot mode is on the delay must be increased sufficiently to allow for this.

Line 70 enable interrupts on receiving a Service Request from ORION.




Lines 75-80 Convert an input character ; to , for output to ORION.

**Program Lines 85 to 110:** send predefined command strings to ORION, to set up channel and task data, followed by the run command to trigger the task.

```
CH 1-20 SE 110

TA 1 OP MI TR TI DE 5 CO* RE IN IN5

TA1 CH1-20 AT F LO EV FO CO MA EV TO GP
```

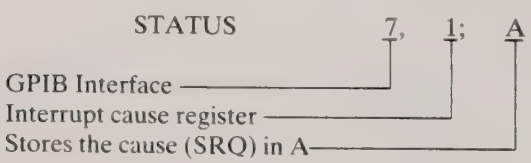
**Program Lines 120 to 140:** forms the control loop round which the program will loop whilst waiting for interrupts from the keyboard or ORION.

Lines 130 checks for depression of the function key 1 at the keyboard and jumps to the "Send command rountine" at line 300, when detected.

Lines 140 checks for depression of function key 4, and jumps to the program termination routine at 390. In addition it displays the function key labels on the screen.

**Program Lines 200 to 280:** the service routine to which the program branches on receiving an ORION interrupt.

Line 210 checks the interrupt cause and clears the interrupt.



Line 220 conducts a serial poll of ORION and stores the status byte contents. (See Note 1).

```
S = SPOLL (701)
```

Line 250 reads the data input from ORION.

Line 260 displays the data on the screen.

Line 270 the interrupts are re-enabled before returning to the control loop.

**Program lines 300 to 370:** the routine that sends a line of command data to ORION.
Lines 330-350 display the prompt for command data and accepts the input from the HP-85 keyboard, which is output to command ORION.

Lines 360 re-establishes interrupts before returning to the control loop.

**Program lines 390 to 410:** the program termination routine, clears the screen, prints 'program terminated', and halts.

**Note 1:** The Status byte, stored in 'S' by the command

```
S = SPOLL (701)
```

can be interrogated to establish the service request type. The SRQ bit, bit 6 of the SRQ byte, will always be set when a SRQ is received, this should therefore be taken into account when checking the byte for request type. When the service request type has been established the ORION data may be processed accordingly.

For example: The service routine, at lines 200 to 280, could be modified to perform processing according to the type of input received from ORION. Change the following lines to output all messages to the HP integral printer and all other data to the VDU.

```
230    IF S=70 THEN GOTO 420
420    !PRINT MESSAGES
430    ENTER 701;B$
440    PRINT B$
450    GOTO 270
```

When the program is run all ORION Messags, eg power fail, power back, program 'WRITE' messages etc., that have been directed to the GPIB interface will be output on the HP-85 printer.

Solartron Instruments

MP MASTER
MA20217
SOLARTRON
Schlumberger